NASA Technical Memorandum 101025

TRENDS: The Aeronautical Post-Test Database Management System

W. S. Bjorkman and M. J. Bondi

January 1990

(NASA-TM-101025) TRENDS: THE AERONAUTICAL POST-TEST DATABASE MANAGEMENT SYSTEM (NASA) 130 p CSCL 14B

N90-25149

Unclas
G3/09 0269464

National Aeronautics and
Space Administration

NASA Technical Memorandum 101025

TRENDS: The Aeronautical Post-Test Database Management System

W. S. Bjorkman, Analytical Mechanics Associates, Inc., Mountain View, California
M. J. Bondi, Ames Research Center, Moffett Field, California

January 1990

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TRENDS: THE AERONAUTICAL POST-TEST DATABASE MANAGEMENT SYSTEM

Table of Contents

Section I
SUMMARY AND INTRODUCTION

Section II
TRENDS OVERVIEW

PRECEDING PAGE BLANK NOT FILMED

Section III
MAJOR HIGHLIGHTS OF TRENDS

Section IV
DATABASE MANAGEMENT

Section V

TRENDS DESIGN/DEVELOPMENT CONSIDERATIONS

Section VI
SUMMARY AND CONCLUSIONS

TRENDS, THE AERONAUTICAL POST-TEST DATABASE SYSTEM

Section I

SUMMARY AND INTRODUCTION

1.0 SUMMARY

This report describes TRENDS, an engineering database operating system developed by NASA to support rotorcraft flight tests. The history, concept, structure, usage, and features of the system are discussed. The purpose of this report is not to give instruction in the use of TRENDS (user manuals are available for that purpose), but rather to document the system's capabilities, structure, features and reasons for being. The report shows that this system is able to support most aeronautical database test and analysis requirements including rotorcraft, fixed-wing aircraft, wind tunnel data, and simulation.

1.1 INTRODUCTION

TRENDS is an interactive Database Operating System developed by NASA to support rotorcraft research studies for NASA and for other government and nongovernment agencies. TRENDS offers a major breakthrough as an engineering database management system for rotorcraft research groups, because it can service both project management and engineering personnel through the use of both narrative and numerical data access and analysis. TRENDS could be considered as the LOTUS 1-2-3 for the aeronautical engineer who would like to do research and data analysis on rotorcraft. The system is fast, powerful, robust, and user-friendly, and has a multidisciplined user interface which allows the user to easily access and plot the data.

This report is written primarily for someone considering TRENDS as a solution to the problem of storing and accessing engineering test data. This does not exclude those interested users or database managers who are already using TRENDS. The report attempts to describe what TRENDS is and what it can do. TRENDS offers the Database Manager a database operating system which may fully meet his needs or may serve as the basis for one which could be developed with only minor modifications.

The acronym TRENDS was derived from Tilt Rotor Engineering Database System because the system was originally developed (beginning about 1982) to support flight testing of the XV-15 tilt-rotor aircraft. The system has been extended to support flight and wind tunnel tests of other rotorcraft, but the name is still appropriate to the system's function and has been retained. Before TRENDS, NASA engineers had only limited capabilities for displaying XV-15 data, and could not cope with the large volume of test data. TRENDS was developed to meet the needs of the engineers and coordinate the data. Appendix A contains a chronology of TRENDS' development.

1.1.1 Users

TRENDS is primarily built as a tool for the nonprogramming aeronautical engineer, but it is also used by individuals of other disciplines with or without computer backgrounds. The system supports a wide variety of engineering disciplines from rotorcraft performance and handling qualities, aeroelastics, dynamics, flight control, and loads to tilt-rotor transitions and simulations. Narrative data complement the numerical data, identifying data items and databased flight segments. The system is designed to provide all of the project information a user needs without having to contact the flight-test engineer. Users can access any of the multiple TRENDS databases with the same software, as shown in figure 1.1. TRENDS currently runs on a VAX computer at Ames; hence, it has all of the advantages of the DEC/VMS operating system, including full networking and remote user support.

New 4th-level programming languages can nowadays provide general capabilities for the creation of computerized databases and displays, but in spite of their generality, they fail to provide what is needed by engineers -- a fully pre-defined engineering interface to the data. These new languages still require the research engineer to configure software calls to obtain the data displays he wants, thereby providing user flexibility, but also requiring him to be a programmer and systems designer. The aeronautical engineer who is interested in viewing and analyzing test data is seldom interested in programming. TRENDS was developed to provide the user with a complete system without requiring him to write software or to locate and mount data tapes.

1.1.2 Features

TRENDS has been configured to be a complete database system which serves a variety of users. Among the features which make TRENDS useful are:

1. Capabilities for multiple users and multiple databases.
2. A flexible, prompt-driven interface with easy defaults.
3. Capabilities for searching and plotting statistical data.
4. Narrative storage and searching features.
5. Support for many different graphics devices.
6. Flexible and capable time-history plotting.
7. Pseudo-flight creation.
8. In-line formula specification and evaluation.
9. Built-in analysis capabilities.
10. Database installation and error-checking tools.

These features will be described in some detail in the following sections of this report.

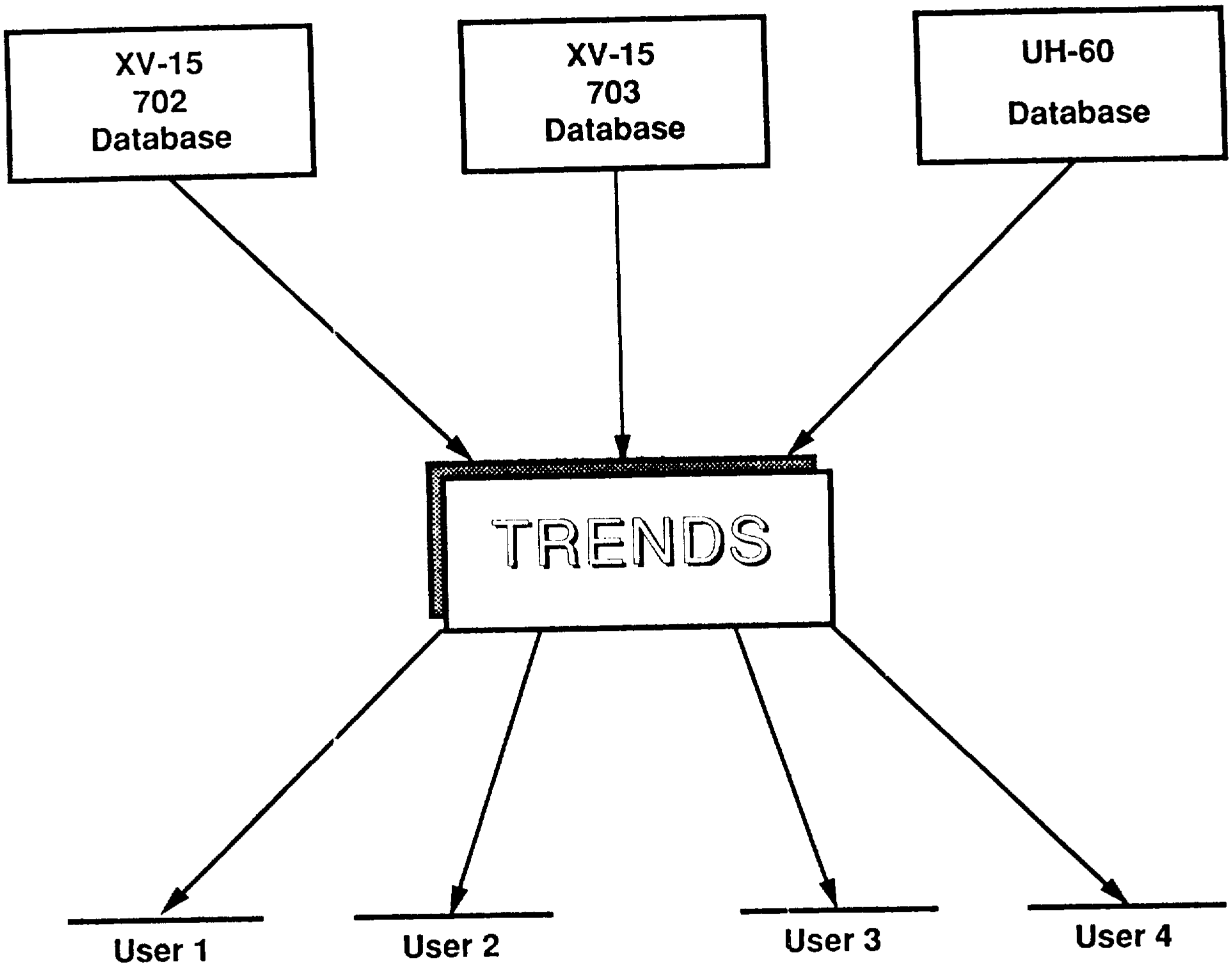


Figure 1.1 Basic TRENDS
Multiple Users of Multiple Databases.

1.2 REPORT STRUCTURE

The structure of the remainder of this report is as follows:

Section 2: TRENDS OVERVIEW
Section 3: TRENDS HIGHLIGHTS
Section 4: DATABASE MANAGEMENT
Section 5: TRENDS DEVELOPMENT
Section 6: CONCLUSIONS

Note: The conclusions presented in Section 6 include a list and discussions of the lessons we have learned as well as our plans for the future expansion of TRENDS.

Section II

TRENDS OVERVIEW

2.0 TRENDS OVERVIEW

This section discusses the TRENDS concept and the considerations which have shaped the system. An overview of the TRENDS database structure is presented. A summary of the TRENDS capabilities is shown in Figure 2.1.

2.1 TRENDS CONCEPT

The initial requirements were to provide a software system to access flight data via a search option and to be able to display statistical min/max data graphically and in tabular printed form. The final product was an interactive, full-blown, menu-driven database operating system. The uniqueness of the concept was that the database was to be immediately accessible to the rotorcraft engineering community and not just to the flight-test engineer who, in the past, would have been responsible for disseminating what he or she concluded were the salient results of the flight test some time (perhaps years) later.

2.1.1 Interactive

The interactive nature of TRENDS is key to the system. The user must be able to interrogate the database directly rather than have to submit a batch job or wait for a data-processing person to generate it. Batch action has a purpose, as does the delegation of tasks to support personnel, but only by hands-on interaction can the serious user most efficiently exercise the tools and the database to solve his problems. On the other hand, the interaction must be simple, robust, and rapid so that senior engineers and managers (or any other users) don't get frustrated by complicated and arcane commands and slow response. The emphasis in the design and development of TRENDS has been to provide an interactive tool which is not only capable, but which can be easily learned and easily used, and which is robust and efficient.

2.1.2 On-line Database

TRENDS was designed to utilize an on-line database in the interests of interaction and efficiency. That is, the information is always accessible, with no requirements for tape-mounts, disk-mounts, or special programs to be run each time to properly install the data in the database. Users can also get immediate access to the flight-test data without having to go through the effort of devising a structure and filling their own databases.

----- TRENDS CAPABILITIES ------

USAGE:

- MENU DRIVEN
- USER-FRIENDLY
- QUICK RESPONSE
- INTERACTIVE

MULTIPLE USERS
REMOTE USERS

SELF-CONTAINED HELP
GRAPHIC TERMINAL SUPPORT

USER PSEUDO FLIGHT GENERATION -------> FROM:

DATA SEARCHING
NARRATIVE SEARCHING
DATABASE SEARCHING

DATABASES: - - - - - - - - - > ON-LINE < - - - - - - - - - - - - - - -

MULTIPLE FLIGHTS
MULTIPLE ROTORCRAFT
---> XV-15(702 & 703)
---> UH-60

MULTIPLE DATABASE TYPES
--> FLIGHT, WINDTUNNEL, MATH MODEL
MULTIPLE DATA TYPES
-> TIME HISTORY, MIN/MAX, NARRATIVE

PLOTTING TYPES

TIME HISTORY
MULTI-FAMILY
MULTI-PLOTS/Page
SNAPSHOTS

AMPLITUDE SPECTRA
HISTOGRAM
MULTI-TEST PTS/Page
DATABASE COMPARISON

MIN/MAX (Statistical)
HARMONICS vs MIN/MAX
CROSS-PLOTTING

PLOTTING ATTRIBUTES ---> user friendliness ---> automatic labeling

Easy to change Scales & Labels
Easy Filtering & Storing of data
Easy Access to Help Features

Easy Plotting of User Defined Functions
Easy Plotting from 1 --> 16 plots/page
Easy Editing/Saving/Recalling of Plot Setups

Transparent cross plotting of parameters having different data rates

PRINTOUTS & VIEWING

STATISTICS (EACH PARAMETER)
DERIVED PARAMETERS
USER DEFINED FORMATS
FUNCTIONS (USER DEFINED)

STATISTICS - KEY PARAMETER GROUPS
PARAMETER DEFINITIONS/PARAMETER GROUPS
HARMONICS
ALL NARRATIVE, NUMERICAL & HELP DATA

DATA SCANNING/SEARCHING ------> Iteratively

FLIGHT LOGS
PROJECT LOGS
FLIGHT DESCRIPTIONS

NUMERICAL DATA (Multi-Parameter)
NARRATIVE DATA (Multi test points)
PARAMETER NAMES/DEFINITIONS

ANALYSIS/In-line Analysis Tools:

FORMULA EVALUATION
REGRESSION
CONVOLUTION FILTER
CYCLE AVERAGING

DERIVATIVE & INTEGRALS EVALUATION
FAST FOURIER TRANSFORMS(FFT)
SIMULATION/GTRSIM/Gateway
DATAMAP/Gateway

Figure 2.1 TRENDS CAPABILITIES

The developers of TRENDS feel that nowadays the only data that will ever be looked at by most users are the data that are on line. The ease with which archived data can be accessed, manipulated, plotted and analyzed during an interactive session has caused users to ask for new TRENDS databases to be established for old test data residing on tapes or disks in various formats.

2.1.3 Multi-Rotorcraft

TRENDS has the ability to move from database to database (e.g., from XV-15 to UH-60 to BV-360 windtunnel data) within a single working session. The commands and tools are the same (or very similar), although names of variables change and counters are numbered differently This feature gives the potential for comparisons.

2.1.4 User-friendliness

User-friendliness is difficult to define explicitly, but easy to spot when it is absent. Great concern has been given to the user-friendliness of TRENDS during design and development. Each change to the user interface is considered carefully as to the logic of prompts, the user's workload, the frustration it is likely to cause, and the input mistakes which may have to be handled. A main menu that shows all of the main options as logical functions is felt to be a good first step toward user-friendliness. Descriptions and supporting narrative to accompany the numerical data are also user-friendly.

2.1.4.1 Main Menu Layout. A concise, well-planned visible menu, in conjunction with patterned prompts and on-line help, is a requirement for user-friendliness. In the TRENDS menu, all major choices available to the user are presented at once without having to branch into an unknown tree type of menu. Brief or detailed on-line help is easily available for each menu item. The TRENDS main menu is column-oriented by logical function. The six column headers, which categorize the listed menu items are:

CONTROL-----> database selection; terminal type; hardcopy option; user functions available; exit from TRENDS

DESCRIPTIVE-> project description; database summary; flight log; flight descriptions; search on narrative; calibration data

DATA-SCAN --> search on numerical data; show key-parameter data; view parameter statistics; customized printout; locate available time-history data

PLOTTING ---> time-history plotting; group plots of performance parameters; plotting of statistical data; time-history plotting of one parameter for many counters; families of plots for statistics

ANALYSIS ---> gateway to analysis programs DATAMAP and GTRSIM; plotting and printing of harmonic amplitudes; derivation and plotting of amplitude spectra; histogram plotting of loads data;

USAGE ------> help; parameter definitions; derived parameters; user-generated file treatment; time-history groups

The order of features within the menu is a matter of concern and is frequently reviewed. Patterned dialog is used to effect the user-computer interface once an option has been selected from the main menu. Because not all users prefer the same option names or orders within the menu or the same dialog, certain compromises have been incorporated to please a broad a range of users.

The combination of a menu with prompting inside the menu items is considered by the authors to be the best user-friendly user-interface format. Forms or pull-down menus and icons have aesthetic appeal, but require very specific terminals, mice, joy-sticks, etc. TRENDS serves a wide variety of interface hardware with a standard scrolling menu, but also provides a screen-managed version of its menu for particular terminals to enable menu-item selection by means of keyboard arrows.

2.1.4.2 Supporting Narrative. A database operating system for test data must include narrative to support the numerical data if is to be considered user-friendly. TRENDS includes data item descriptions, flight and test descriptions and project information among its available narrative data. Test descriptions are related to numerical data by flight and test-point numbers.

2.1.4.3 Plot-specification Dialogue. One of the most user-friendly features of TRENDS is the simple-yet-powerful, flexible, plot-specification dialogue. The user may specify as little as the names of the parameters used for abscissa and ordinate. TRENDS will then provide scales, labels, headers and plots. The user may, alternatively, specify everything through a straightforward, logical entry syntax. On-line help is available at every step of the dialogue.

2.1.4.4 Input Error Checking. It is not necessary to crash the system to find whether a user's input is proper or if a parameter exists, provided the software performs suitable checks. TRENDS checks every character input by the user which could be either a parameter or a function of parameters and determines whether there are any illegal entries. Certain checks are made immediately to determine whether requested data exist in the database. In the name of user-friendliness we are able to eliminate many crashes for incorrect parameter names which might crash the system and frustrate the user if not screened out.

2.1.4.5 On-line Analysis. TRENDS lets the user specify formulas involving stored parameters and evaluates these formulas immediately for plotting, searching or listing. In addition, many standard analyses are provided within the system. It is not necessary to program other software or to leave TRENDS in order to filter the data, fit the data to polynomials or perform spectral analyses.

2.1.5 System Speed

Great concern has been given to computational efficiency. The chosen VAX-specific file structure enables direct (i.e., keyed) access to the information of interest without extra searching or data-reading. Opening of extra files has been regarded as a time-consuming overhead item and has been held to a minimum. The result of the effort and concern is a system which can search and display large databases rapidly, thereby minimizing the user's effort and frustration.

2.1.6 User Software Exclusion

Potential users of the data often ask how to get the archived data into their own favorite analysis programs or plotting packages. Modules have been supplied for some users to take data, but this is not standard. The choice was made to provide the tools most needed (e.g., plotting routines, spectral analysis) along with the data, so that engineers don't have to devise, interface, and check out tools to do their jobs. This method of operation has been happily accepted by most of the engineering users of TRENDS. Additional requested features and capabilities have been developed and incorporated into the system when they have been judged to have general interest or application. A provision has also been made to access certain other software tools (i.e., DATAMAP and GTRSIM) through TRENDS and to enable them to use data from TRENDS' databases.

2.1.7 Documentation

Users can usually find all the help they need in TRENDS during a working session. User help includes brief descriptions of each menu item and general hints on the use of the system. It also includes detailed help and examples for each program feature. The user can also call up information concerning the database, the project, and the stored variables as part of the normal access of narrative data.

In addition TRENDS is supported by several manuals:

1. TRENDS User's Manual
2. TRENDS User's Reference
3. TRENDS Procedures Manual

Copies of these manuals may be obtained by calling Mike Bondi at NASA Ames, code FAF, (415) 604-6341 or Bill Bjorkman at AMA (415) 964-1844.

2.2 TRENDS DATABASES

Engineering tests, be they flight tests, wind tunnel tests, simulations, or any other tests during which experimental data are collected, have common characteristics. The way that TRENDS accommodates test data from multiple sources, multiple sensors, and multiple test points is to

1. Assign a separate directory for the test data for each vehicle (e.g., segregate XV-15 data from UH-60 data)
2. Assign a unique index to each test point
3. Assign a mnemonic to each sensor and store all of the data for that sensor in a file named for the mnemonic (exceptions will be discussed later)

The path to a test data point is thus given by specifying the test vehicle (which leads to the directory), the sensor mnemonic (which leads to the file), and the test-point index (which leads to the record of interest).

2.2.1 Test Points

The test-point index is commonly called a "counter" in TRENDS because there is a physical counting device in the XV-15 by which the pilot increments the counter for each new test point. This counter is recorded with the other test-point data. XV-15 counters are not reset between flights and thus continue to increase. The counter indices for XV-15 aircraft 703 are now around 14500 for flight 234 (they are also incremented during ground and hangar runs). Counters for UH-60 tests are constructed from flight and run numbers to create a unique test-point index, since run numbers are reset for each flight. Wind tunnel counters are also constructed to give the uniqueness TRENDS requires for rapid record access. The term counter is often used to mean the test-point event as well as the index of that event. Thus "counter duration" means the recorded length of the test-point event.

Each test point of a well-planned engineering test project serves a planned purpose. That purpose or the specific conditions for the test point can usually be described in a few key words of narrative. TRENDS stores and enables a search on these short test-point descriptions as part of the database information. This description, together with recorded and derived data from all of the sensors, completes the information stored for each test point and is indexed (i.e., related and located) by the counter number.

2.2.2 Data Types

Most of the numerical data recorded during engineering tests is in the form of time-histories which consist of a time-series of samples of the measured output from each sensor. Many of the rotorcraft test points are taken at a trimmed or steady condition where many time-histories are basically constant and where the mean values of such parameters are just as useful as the time-histories and are much more economically stored. Other statistics besides the mean are also useful (especially per-rev statistics for rotorcraft), so TRENDS precomputes some of them. Such statistical measures (some of which are listed in paragraph 3.2.5) are sometimes called Min/Max data. In the case of wind tunnel data, time histories are usually not available and only the mean values are recorded. As noted earlier, a TRENDS database is completed by adding narrative information in support of the numerical data.

2.2.2.1 Min/Max. This terminology came about because of a digitizing process in which only the minimum and maximum values of the time during each rotor revolution were stored as digital values. These were then used to derive single statistical values for each sensor over the time span of the counter for (1) the average "steady" value over all revs in the counter, (2) the average oscillatory value, and (3) the maximum oscillatory value.

The steady value for one rev is the average of the minimum and maximum values, while the oscillatory value is half of the difference between the maximum and minimum values. Such measures are only meaningful for an engineering test which is influenced by a cyclic phenomenon such as a turning rotor, but statistical measures of some kind (at least the mean value) serve well in any engineering database system to describe the "big picture" covering a collection of different test points. The term "Min/Max" is often used to refer to all such statistical measures in TRENDS.

TRENDS stores a set of statistical measures (Min/Max data) for each parameter and counter of each database. The particular statistics are customized for the particular database (see paragraph 3.2.5). TRENDS has many features for searching, listing, and plotting Min/Max data.

2.2.2.2 Time-History. TRENDS accommodates parameters collected at different data rates with total user transparency when plotting and/or cross-plotting. The requirements for storage and use of time-history data differ between sensors or parameters. For example, altitude or air-speed data are of interest only at a low sampling rate, but must span the duration of the counter. On the other hand, bending moment data for a blade element must be sampled at a high rate to be useful, but need not necessarily span the full counter duration. TRENDS treats several types of time-history data and several "groups". Groups combine several test-data parameters of similar type (e.g., loads, performance, aeroelastics). Some types of time-history data are filtered and decimated before being stored in the database. Not all groups of time-history data are stored for a given counter because of storage limitations. The decision of whether to store certain groups is made by project engineers, as is the decision about appropriate filtering and decimation rates.

2.2.2.3 Narrative. The numerical information stored in a TRENDS database is supported by extensive narrative (textual) information, including

1. Test-point descriptions
2. Detailed flight descriptions
3. Flight logs (brief descriptions)
4. Project data on the aircraft and test program
5. Parameter (data-item) descriptions
6. On-line help for the user

The test-point, flight, and parameter descriptions may be searched by the user. That is, the user can specify character strings (words or phrases) to be used in locating certain stored information. The set of related test points resulting from a text search may be stored and used to identify the data region for subsequent plots or searches. Stored narrative information for flights and counters is used for automatic labeling and headers on plots.

2.2.3 Database Management

The tasks of database management are not given to the general user, according to the TRENDS operational philosophy, but to a database manager. The manager is responsible for filling and editing the database and for monitoring the quality of the data. The manager must find enough disk space, reduce and store the data, assess and fix or report bad data, and process requests from users. Most database managers to date have been support personnel, but selected engineers were given the responsibility for managing the UH-60 flight-test database.

Menus are provided for the database managers to integrate the processes and help in the filling and editing tasks. Supporting files are updated automatically in most cases as part of the filling process. Programs for assisting in the structuring and entry of narrative data are provided as part of the management software. Database management considerations will be discussed in Section IV.

2.3 DATABASES AT AMES

Several rotorcraft test and simulation databases are maintained at Ames. Many similarities may be observed among them. They are all characterized by having multiple parameters and data regions (i.e., runs or test points) and they all have some narrative descriptions to support the numerical results. Databases differ, however, because not all objectives are the same and not all instrumentation systems and data-reduction systems are the same. Each of the current databases at Ames has its own unique aspects.

2.3.1 XV-15 Database

TRENDS was developed to treat XV-15 data; hence, most of the current storage is devoted to the XV-15. Among the unique aspects of the XV-15 database are the following:

1. The counter sequence is not reset for each flight.
2. Four-character itemcodes are used to identify the parameters.
3. Time-history data are categorized into groups.
4. Slopes are calculated and stored for some items.
5. Calibrations are included in the database.
6. There are two XV-15 databases, one for each aircraft.

2.3.2 UH-60 Database

TRENDS was used to support the Phase I NASA/Army flight test of the highly instrumented UH-60 at AEFA and has become a basic tool of the flight-test engineers associated with the project. Some unique aspects of the UH-60 database are the following:

1. Appointed user-engineers, not support personnel, are the Database Managers.
2. Counters are composed of flight and run numbers.
3. Both mnemonics and itemcodes identify data items.
4. Three databases, representing different data releases, are currently accessible.

2.3.3 Wind Tunnel Databases

The TRENDS databases at Ames include wind tunnel databases for HARP and BV360. These were built from input formatted disk files made from data recorded at the DNW facility in The Netherlands. Counter numbers were derived from non-numeric run numbers. "Flight" numbers were derived from the test date. Only run descriptions and mean values of parameters are stored in the database because neither time-histories nor statistics other than the mean were recorded.

2.3.4 Simulation

Results from the tilt-rotor mathematical model simulation, GTRSIM, (see Section 3.4.2) are stored in the user's directory as a database which may be accessed by the GTRSIM/TRENDS interface software for listing, plotting, or comparisons. Counter numbers for simulation runs are supplied by the user or, if inputs to GTRSIM have been taken from the flight-test database, are automatically supplied (as the flight-test counter number). The form in which simulation results are stored differs from the form used in the aeronautical test databases. Future efforts will change the form of the database for simulation results to bring it into agreement with the flight-test databases.

Section III
MAJOR HIGHLIGHTS OF TRENDS

3.0 HIGHLIGHTS SYNOPSIS

This section presents the major highlights of the use of TRENDS to access engineering test databases. These highlights are the menu, the plotting features, the searching capabilities, and the analysis tools. This presentation is not an exhaustive compilation of the capabilities of the system, but rather a representative demonstration (mostly by means of examples) of some of its most powerful and useful features. Many of the system's good features are subtle, user-friendly implementations which can be appreciated only by using the system.

Not all of the capabilities of TRENDS will be addressed here, but the menu and the plotting and searching features and the analysis tools will each be discussed in some detail because these features illustrate the full power of TRENDS.

3.1 TRENDS MENU

The layout and presentation of the main menu have been carefully designed to provide access to the various capabilities of TRENDS in a logical and easy-to-use way. The main menu of TRENDS is shown below.

```
                                  ***  TRENDS MENU  ***
    Control        Descriptive Data-scan     Plotting     Analysis     Usage
    -------        ----------- ---------     --------     --------     -----
703>TAIL NO.       PROJECT     SEARCH        TIMEHIST     DATAMAP      HELP
 GR>TERMINAL       DATABASE    KEYS          PERFPLOT     HARMONIC     ITEMDEFS
 NO>PLTHDCPY       LOGSCAN     VIEW          MINMAX       SPECTRA      DERIVED
    FUNCTION       FLIGHTS     CPRINT        QWIKPLOT     LOADS        FILES
    EXIT           WORDSCAN    FIND          MULTIPLT     SIMULATE     GROUPS
                   CALIBS                    COMPARE
```

The menu shown is for XV-15 databases. The column headings express the general function of the menu items shown below them. This categorization is frequently reviewed to make it as helpful as possible to the user. Each menu item corresponds to a program feature (i.e., program, subroutine, or process) which will, upon invocation, prompt users for further specifications of what they want, through other menus or specific questions. Some minor differences will be seen in the UH-60 menu because some of these menu items are not applicable to the UH-60 database. A program feature is invoked by typing the name of that feature in response to the prompt "YOUR CHOICE :" A user need not type the whole name, but only enough of it to make the choice unique. This ability to abbreviate helps avoid typographical errors and is especially favored by poor typists. When the program feature has been completed or purposely interrupted, TRENDS always returns to this main menu unless the choice is "EXIT".

3.1.1 Brief Description of Menu Items

When the menu item "HELP" is chosen, the following brief descriptions of the menu items and other help-options are shown.

DATABASE ACCESS OPTIONS
****

TAIL NO.	Change aircraft of interest
TERMINAL	Assign new terminal characteristics
PLTHDCPY	Change plot-hardcopy option
FUNCTION	List/verify/edit the defined-function file
EXIT	Exit the program, return to the operating system
PROJECT	Display project and aircraft information
DATABASE	Show a brief summary of flights in the database
LOGSCAN	Scan the flight log and search descriptions
FLIGHTS	Display some or all flight descriptions
WORDSCAN	Scan counter descriptions for words or strings
CALIBS	View calibration data by item and flight
SEARCH	Search for a specific set of flight conditions
KEYS	Show value of primary condition keys for a flight
VIEW	View item statistics for specified counters
CPRINT	PRINT item statistics in your own custom format
FIND	Find counters with data for time-history items
TIMEHIST	Plot time-history or spectral data
PERFPLOT	Plot performance items 2X2, 3X3 or 4X4 per page
MINMAX	Plot Min/Max-per-counter data (statistical summaries)
QWIKPLOT	Quick-plot time-histories for multiple counters
MULTIPLT	Plot families of Min/Max data
COMPARE	Co-plot time-history data for two different databases
DATAMAP	Analyze time-history data with DATAMAP
HARMONIC	Display n-per-rev harmonics vs. Min/Max items
SPECTRA	Spectral analysis amplitude display (0-60 Hz only)
LOADS	Show Min/Max/rev data and loads distribution
SIMULATE	Set up and run the tilt-rotor math model program
HELP	Show this list of keyword descriptions
ITEMDEFS	Show/search itemcodes and definitions
DERIVED	Show the derived pseudo-items
FILES	Scan user-created files
GROUPS	Show the itemcodes in the time-history groups

Further, more-detailed descriptions of each of the program features and their use may be called up by the user.

3.1.2 TRENDS-User Dialogue

Once the user has selected a menu item, TRENDS will prompt for the rest of the information which must be specified to find, print, or plot data from the database. Figure 3.1 illustrates the dialogue and the action taken by TRENDS in producing a printout or plot of Min/Max statistical data. A similar dialogue serves users in specifying time-history data. Figure 3.1 shows that for menu selections VIEW, SEARCH, MINMAX, or MULTIPLT the user would first be prompted for an itemcode. In most cases, a mathematical expression can be specified as well as a simple itemcode at this point. There may be more prompts for itemcode, as in SEARCH, where the user sets up a condition template for defining success. The prompt may not specifically ask for itemcode, as in MINMAX, where the user is actually prompted for abscissae and ordinates to set up the plot format. In any case, having specified the itemcode, the user is prompted for a data region (e.g., the flight number). TRENDS then reads a supporting file, FLTCTRMAP.KEY, to find which counters are in the specified flight, opens the itemcode files (files in the database, containing the Min/Max statistical values for each data item) and reads in the data for the requested counters. Then, if the selected menu item was MINMAX or MULTIPLT, TRENDS proceeds to produce plots; otherwise, the information is displayed in printed form.

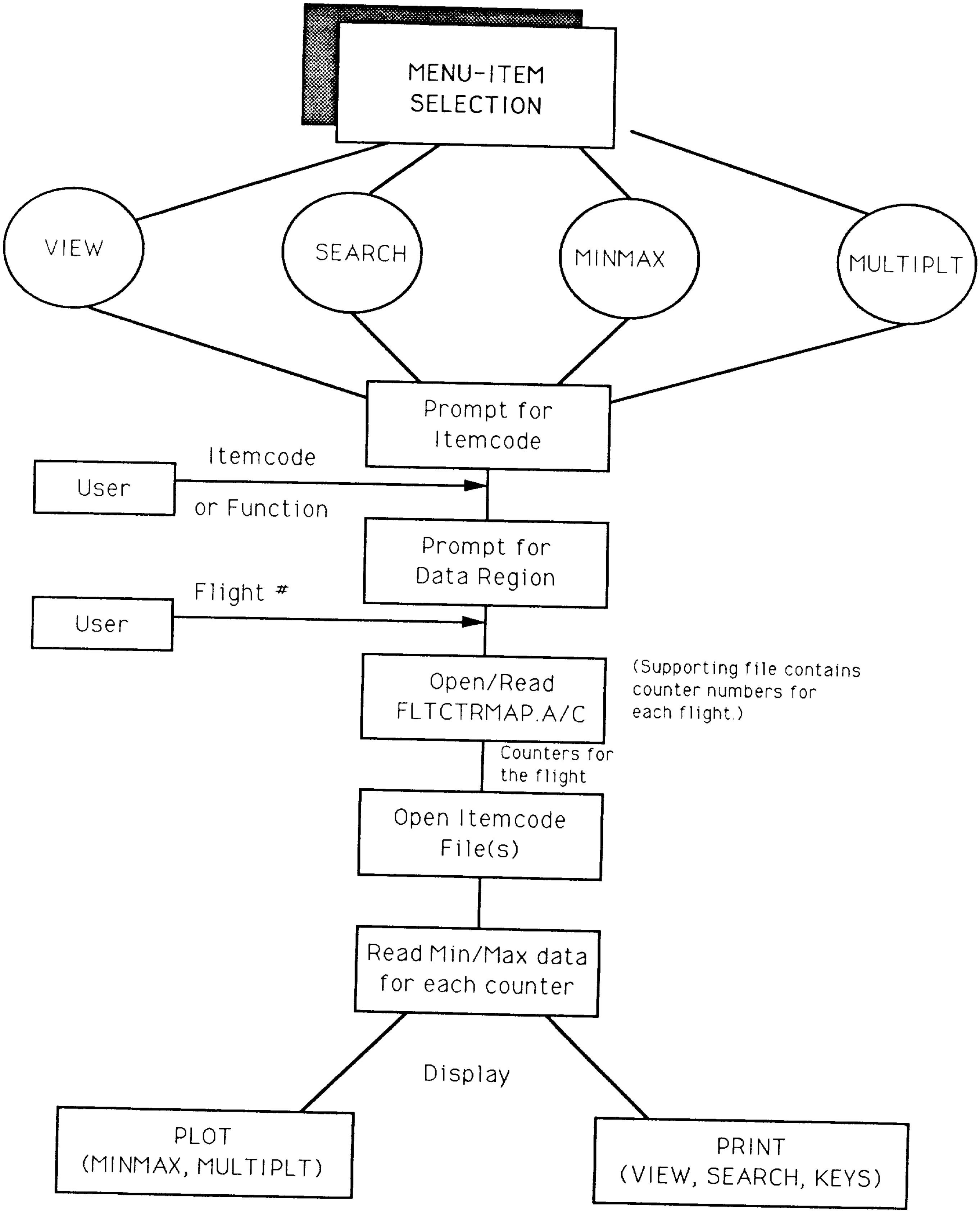


Figure 3.1 Dialog and Action Taken by TRENDS to Produce Min/Max Statistical Data

3.2 PLOTTING

Another highlight of TRENDS is its plotting capability. The best characteristic of TRENDS' plotting implementation is the ease with which a particular plot may be requested and produced. If users want a time-history or Min/Max plot, they need to specify only the abscissa, ordinate, and counter(s). TRENDS will scale and label the axes and write titles automatically. The user may optionally specify the scales, labels, and titles by means of a simple and logical input syntax.

TRENDS' plotting features are not completely general, but still provide considerable structured flexibility. Complete generality requires too much effort from the engineering user who usually just wants to see the data displayed usefully on a well-labeled, well-scaled plot.

The various plotting features which will be described are

1. TIMEHIST: TRENDS' primary time-history display feature
2. PERFPLOT: A time-history snapshot of performance items
3. QWIKPLOT: A "strip-chart" plot for multiple test points
4. SPECTRA: Spectral analysis plots of amplitude vs. frequency
5. MINMAX: Cross-plots of Min/Max statistical measures
6. HARMONIC: Plots of harmonic amplitudes versus Min/Max values
7. LOADS: Frequency distribution histograms of per-rev samples versus load range.

Many different graphics terminals are served by TRENDS. A low-resolution printer-plot capability is available for nongraphic terminals. Hard copies of the plots may be generated in addition to (or instead of) screen plots. The hard-copy plots can be routed to laser or Versatek printers at Ames via dialogue supplied at exit from TRENDS.

3.2.1 Time-History Data

Several menu items or program features may be used to plot time-history data. The most used and most flexible of these is TIMEHIST. It is capable of plotting from one to three plots per page with one or two curves per plot. Each plot page may thus contain from one to six curves. The curves may represent evaluations of formulas involving recorded time-histories. They may also be polynomials resulting from regression computations on the ordinate as a function of the abscissa (which need not be time, but which may also be another formula involving recorded time-histories). The curves may be filtered versions of recorded time-histories or evaluated formulas or a variety of other quantities. The abscissa may be rotor azimuth (derived) as well as a time-history function. Time shifts for the curves may be specified by the user. Examples of time-history plots are shown in figure 3.2.

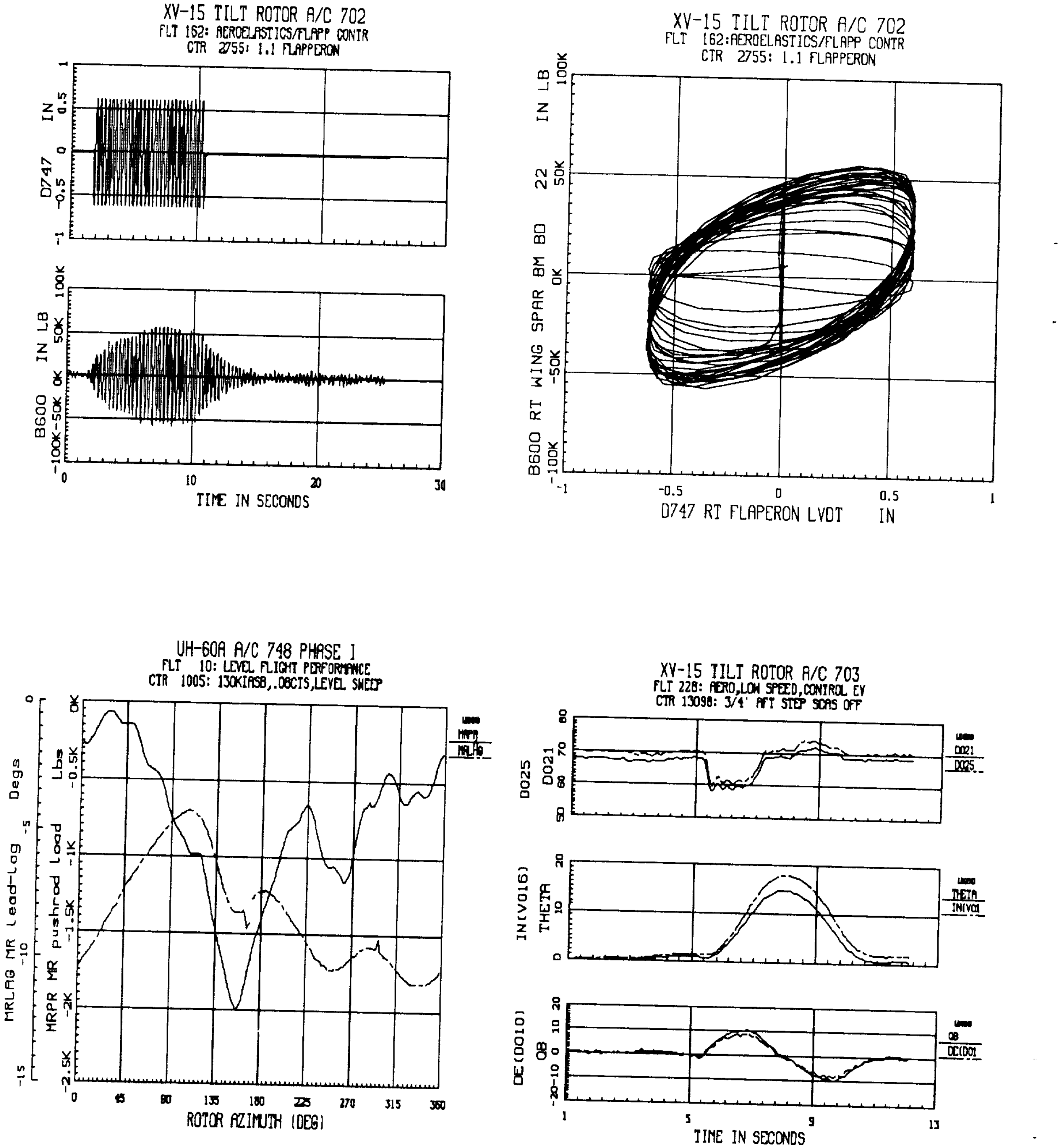


FIGURE 3.2 TIMEHIST plotting examples

3.2.1.1 Functions. The abscissa and ordinates may be specified as formulas as well as simple time-histories. They may also be specified as the names of prestored formulas. The following valid examples show some of the formula-evaluation capability of TRENDS.

	Prompt	Response
	PLOT 1 X-AXIS :	(M143 + M107) / 2
	Y-CURVE 1 :	(M143 - M107) / 2
or -->	Y-CURVE 1 :	POLY(CP*SIN(ALPHA),3)
or -->	Y-CURVE 1 :	-.5 * SFUNC * GTABLE(VT)

These examples show algebraic combination of parameters, a third-order polynomial (POLY) of a formula involving parameters and the use of a stored formula (SFUNC) in a formula involving a table lookup (GTABLE). TRENDS parses the inputs to test their syntactical correctness and disallows the entry if it is improper. This formula-evaluation feature is available for use with Min/Max plots and for searches as well as for time-history data combination. Other available functions include integrals and derivatives and a digital (convolution) filter in combination with formulas. The syntax for formula entry is described in Appendix B

3.2.1.2 Labels and Scales. The examples shown above would all result in automatic scaling and labeling by TRENDS. If users want to specify these, they would enter (for example):

PLOT 1 X-AXIS : AVERAGE TORQUE = (M143+M107)/2,-50000,100000,25000

This specification gives "AVERAGE TORQUE" as the abscissa label and scales the abscissa from -50,000 to +100,000 in grid increments of 25,000. Plot headers (titles) are usually automatic (showing flight and counter descriptions along with aircraft name), but may be overridden completely or in part with user-specified titles.

3.2.1.3 Editing. TRENDS contains an editing feature for recalling and (optionally) changing previously entered plot setups. This feature lets the user save a complex plot specification for later use on another flight or counter. It also lets the user easily change one line (ordinate or abscissa) without having to reenter the rest of the setup specification.

3.2.1.4 Storing. TRENDS has a provision for storing time-histories in the user's directory for later recall. This is particularly useful for storing derived time-histories such as filtered time-series, complex formulas or polynomial regressions. Such stored time-series may be recalled by name for use in formulas. One might store the filtered version of a parameter's time-history, then recall it to use in a formula for the residual (difference) between the "raw" (unfiltered) and filtered representation of the parameter's behavior. This type of storing is different from storing formulas, because the actual data values for each sample point are stored by the counter for which the function is evaluated, while stored formulas are reevaluated from the database and from previously stored time-histories.

Another useful means of storing time-histories is available in TRENDS. This feature permits writing raw or derived time-histories to an output file in ASCII format for transmittal to another computer, such as a PC. This operation is implemented by use of the PRINT command in TIMEHIST.

3.2.1.5 Help. TRENDS provides help menus for all of the various plotting questions one might have while using TIMEHIST or MINMAX (the most-used plotting features in TRENDS). These menus are called out by the user by typing a "?" at any prompt. The help-menu items include provisions for examining the available database entries as well as instructions and examples. The TIMEHIST help-menus are

TIMEHIST SETUP HELP TOPICS

AZIMUTH	COMSCALE	CVF	DERIV	EDIT	EXAMPLES	FORMULAS
FREQ	GENERAL	INTEG	INTERVAL	MATHLIB	MNEMONIC	POLY
PRINT	RECALL	REPEAT	SORT	STORE	SYNTAX	UNSTORE
* or ALL						

TIMEHIST DATA-REGION HELP TOPICS

DATABASE	EDIT	EXAMPLES	FINDCTRS	HARDCOPY	INTERVAL	MYDCS
PRINT	RESCALE	SAVE	SYNTAX	TERMINAL	TITLE	TSHIFT
W80,W132	+(xhair)					

3.2.2 Data Snapshot of Test Point

PERFPLOT provides a means for drawing 4, 9, or 16 time-history plots of different parameters on the same page in a square array. This feature is useful for grouping flight-performance parameters for a visual quick look at a counter. The current implementation permits no formula evaluation or scale or label specification and permits only time as the abscissa. Only one curve is allowed per plot. On the side of flexibility, the user is provided with a capability for specifying which parameters are to be plotted and where they will appear on the page. When one of the specified parameters is not available for the specified counter, its plot space is left blank except for the parameter's name. The user may save plot setups in named files for later recall and modification. Examples of performance plots are shown in figure 3.3.

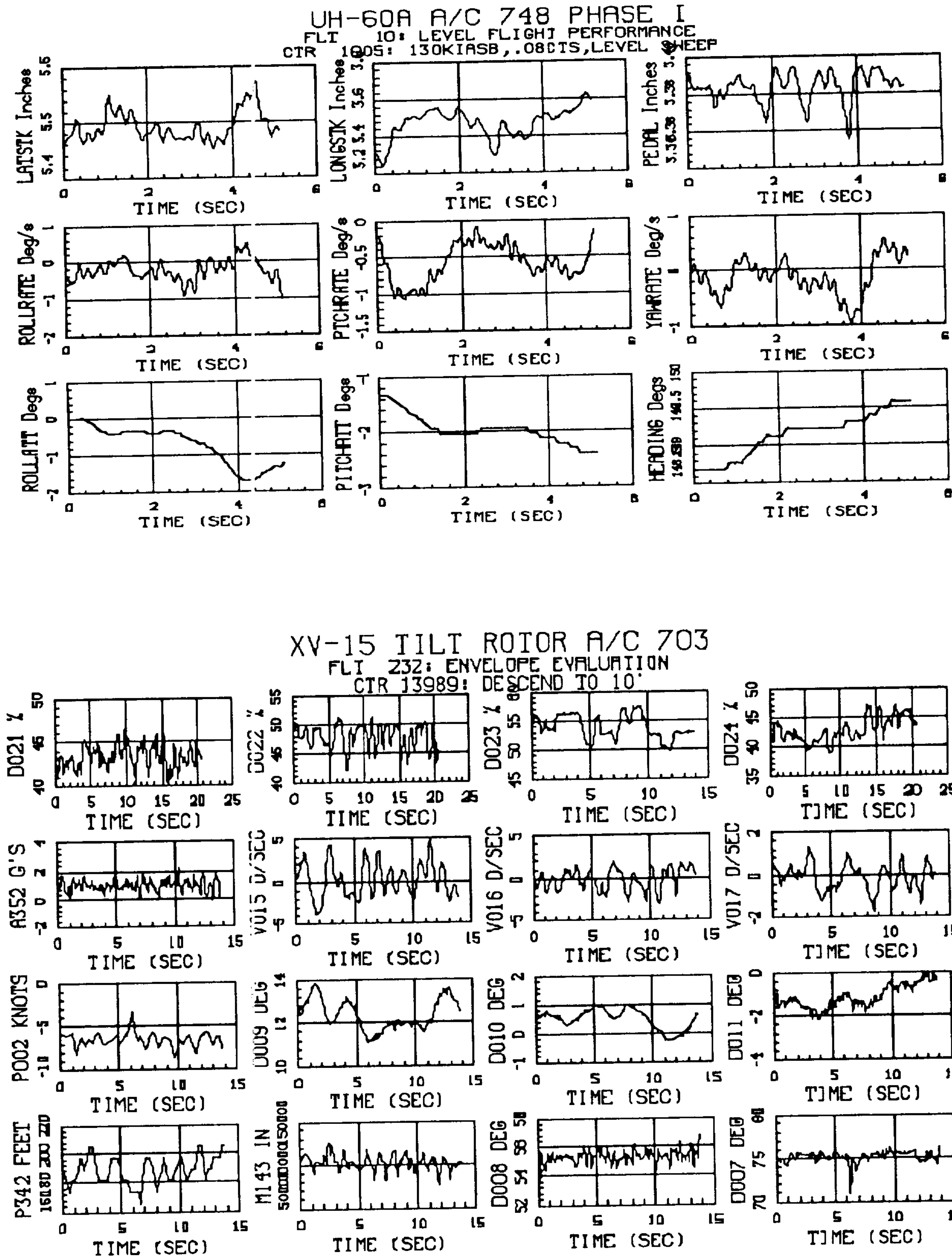


Figure 3.3. Performance Plots

3.2.3 Strip-Chart Plots for Multiple Test Points

QWIKPLOT provides the means for plotting time-histories of a data item's behavior versus time with up to five test points represented per page. The plot setup process is extremely simple, and the special plotting features such as scaling, labeling, and formula evaluation are precluded. TRENDS will take a specified flight or set of counters, divide the counters up into groups of five, and quickly display the strip charts with a time-scale which is common for all plots of a page. This feature gives a "quick look" at the time histories so that the analyst can rapidly scan the data for anomalies or notable behavior for further detailed study. QWIKPLOT examples are shown in figure 3.4.

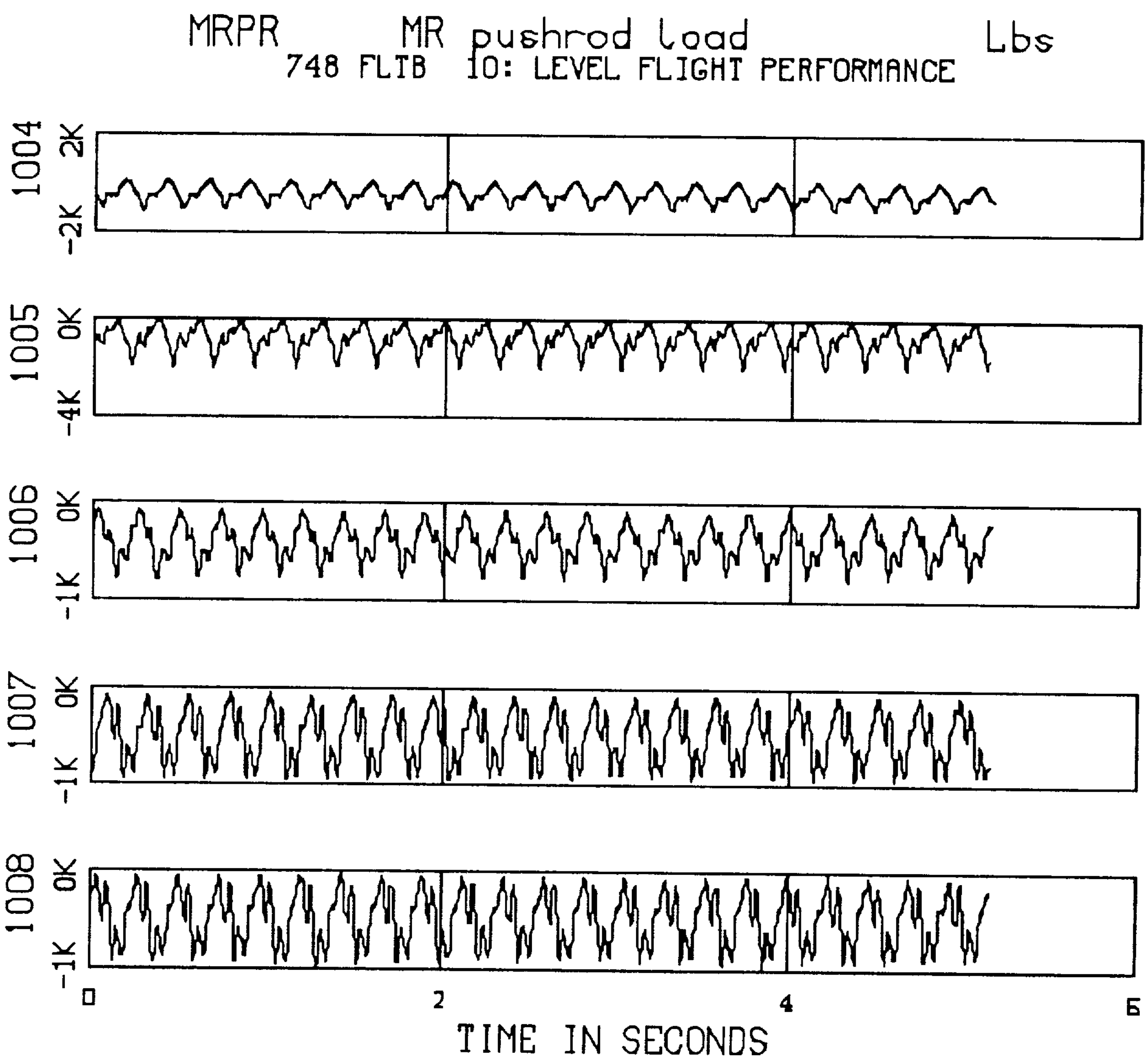


Figure 3.4. QWIKPLOT Example

3.2.4 Frequency Data P ots

TRENDS' analytical capabilities include Fast Fourier Transform (FFT) analyses of spec fied time-histories. The specified time-histories may be formulas involving several stored time-histories or functions of time-histories such as stored filtered functions. TRENDS plots the amplitude spectra versus frequency, one curve per plot, but up to three plots per page. This type of plot may be produced by specifying the abscissa to be FREQ in TIMEHIST or (in simplified form) by using the menu item SPEC̈RA. Examples of frequency data plots are shown in figure 3.5.

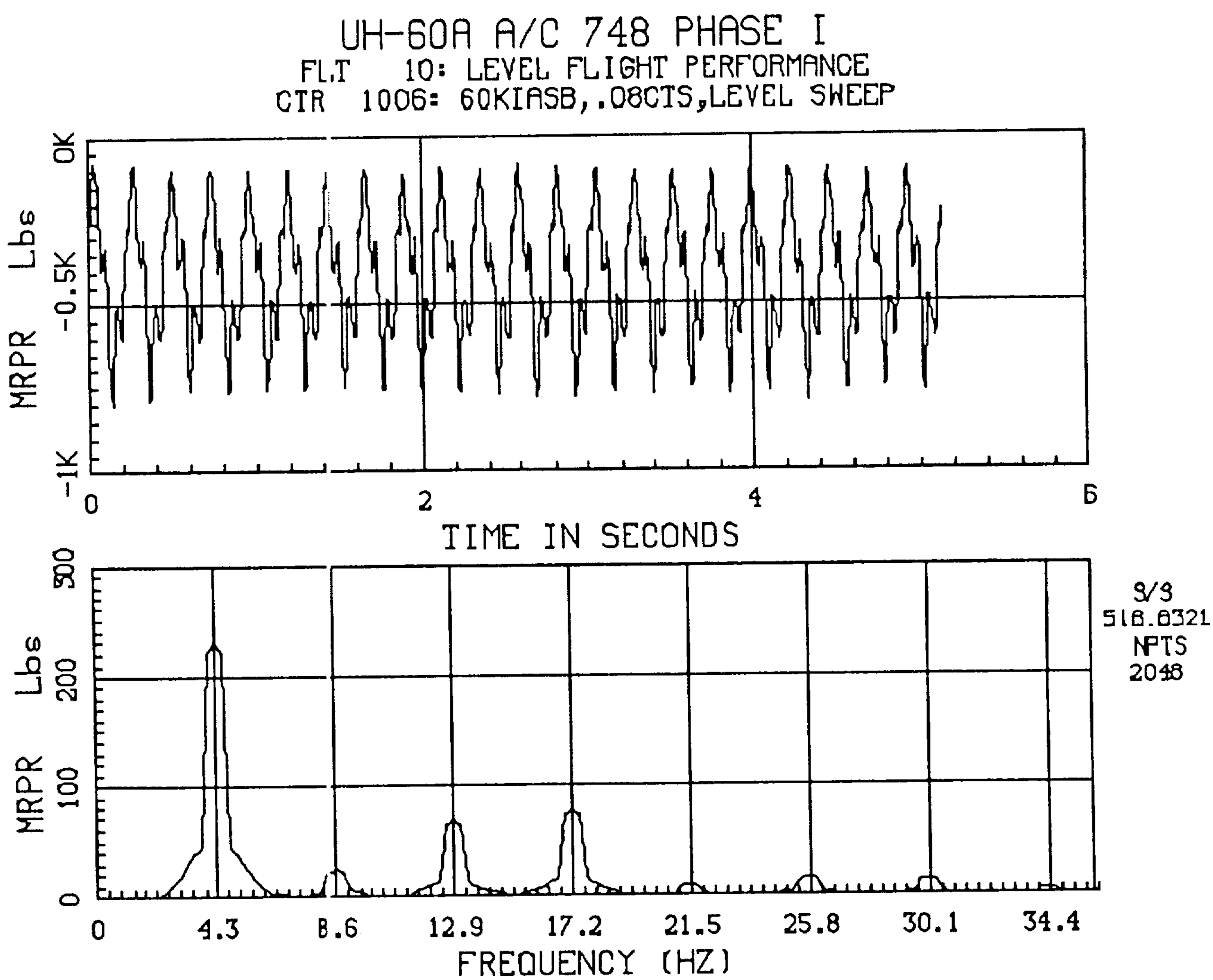


Figure 3.5. Spectral Examples

3.2.5 Statistical Data Plots

Statistical data (Min/Max and harmonic) are plotted by MINMAX, MULTIPLT and HARMONIC. Specification syntax of MINMAX is very similar to that of TIMEHIST and all of the rules for scaling, labeling, and formula evaluation are identical. The data region for one MINMAX plot is a set of counters (e.g., a flight or a pseudo-flight) for which points are to be plotted. As with TIMEHIST, each plot page may contain up to three plots and one or two "curves" of discrete points per plot. MULTIPLT differs from MINMAX in that each MULTIPLT "curve" represents the same ordinate expression (e.g., formula) for a different set of counters while each MINMAX curve represents a different ordinate expression for the same set of counters. HARMONIC plots prestored harmonic amplitudes of data items computed from the one-per-rev versus Min/Max statistics or expressions involving Min/Max statistics. HARMONIC also lists amplitude and phase versus harmonic number.

The prestored statistics for the XV-15 and UH-60 databases are

XV-15 Tilt Rotor

AVS	Average steady (average of per-rev (max+min)/2) DEFAULT
OSC	Average oscillatory (average of per-rev (max-min)/2)
MAX	Maximum oscillatory (max-min)/2 encountered on all revs
SMO	Steady value when maximum oscillatory value occurred
CMN	Algebraic minimum of all raw data samples in the counter
CMX	Algebraic maximum of all raw data samples in the counter
FSC	Full-scale value used (corresponds to 126 byte-counts)
HMn	nth harmonic amplitude (n between 0 and 6, inclusive)

UH-60 Blackhawk

AVG	Average or mean of all of the recorded samples DEFAULT
MAX	Algebraic maximum of all of the recorded samples
MIN	Algebraic minimum of all of the recorded samples
SDEV	Standard deviation of all samples about the mean
AVS	Average steady (average of per-rev (max+min)/2)
AVO	Average oscillatory (average of per-rev (max-min)/2)
V95%	95th-percentile vibratory (95% vibs below this)
MXV	Maximum vibratory (oscillatory) value encountered
SMXV	Steady value when MXV occurred
HMn	nth harmonic amplitude (n between 0 and 15, inclusive)

MINMAX permits the plotting of functions of any of these statistics against functions of any others. Examples of MINMAX, MULTIPLT and HARMONIC plots are shown in figures 3.6, 3.7, and 3.8.

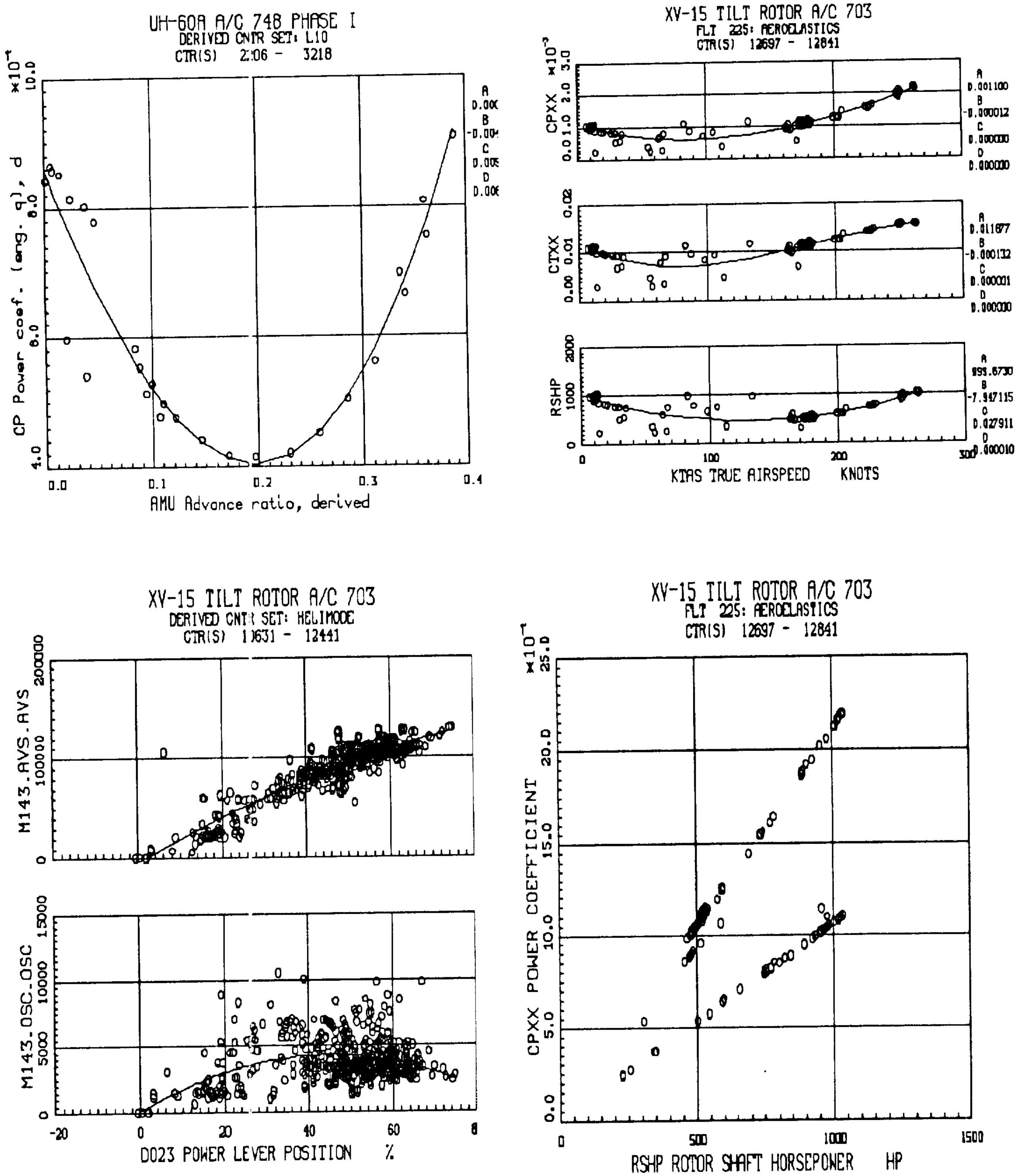
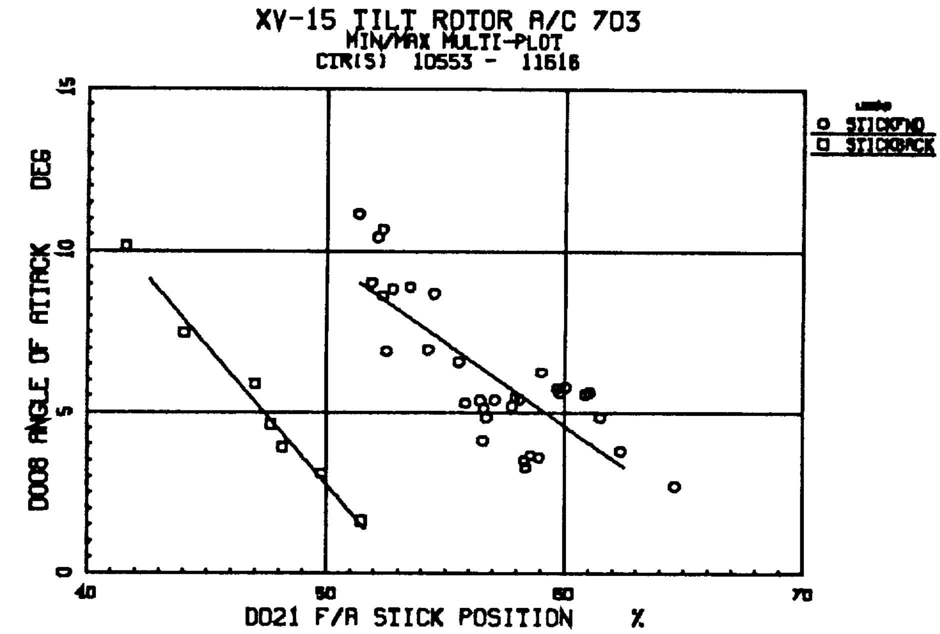


Figure 3.6. MINMAX Plots



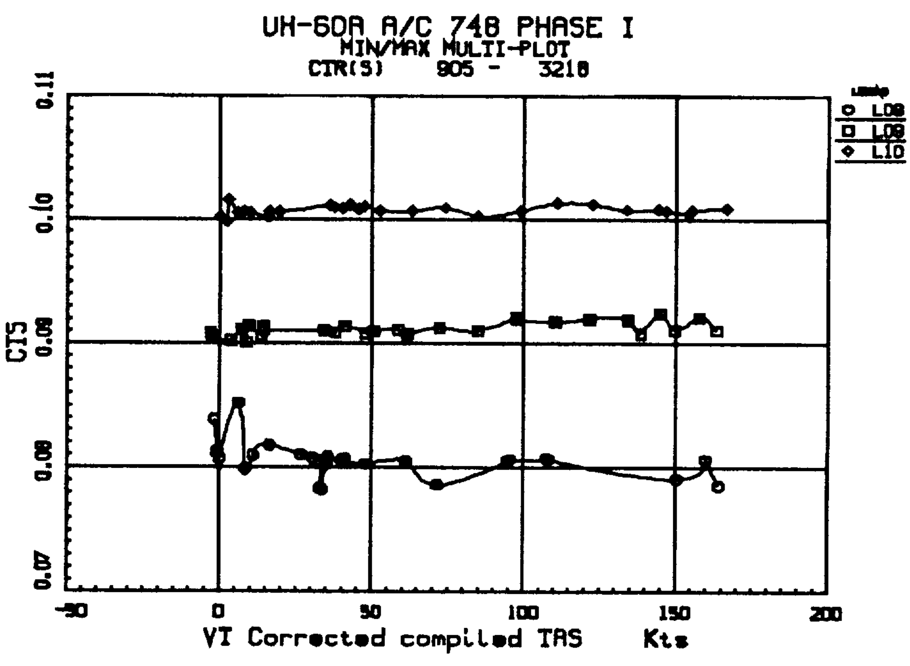
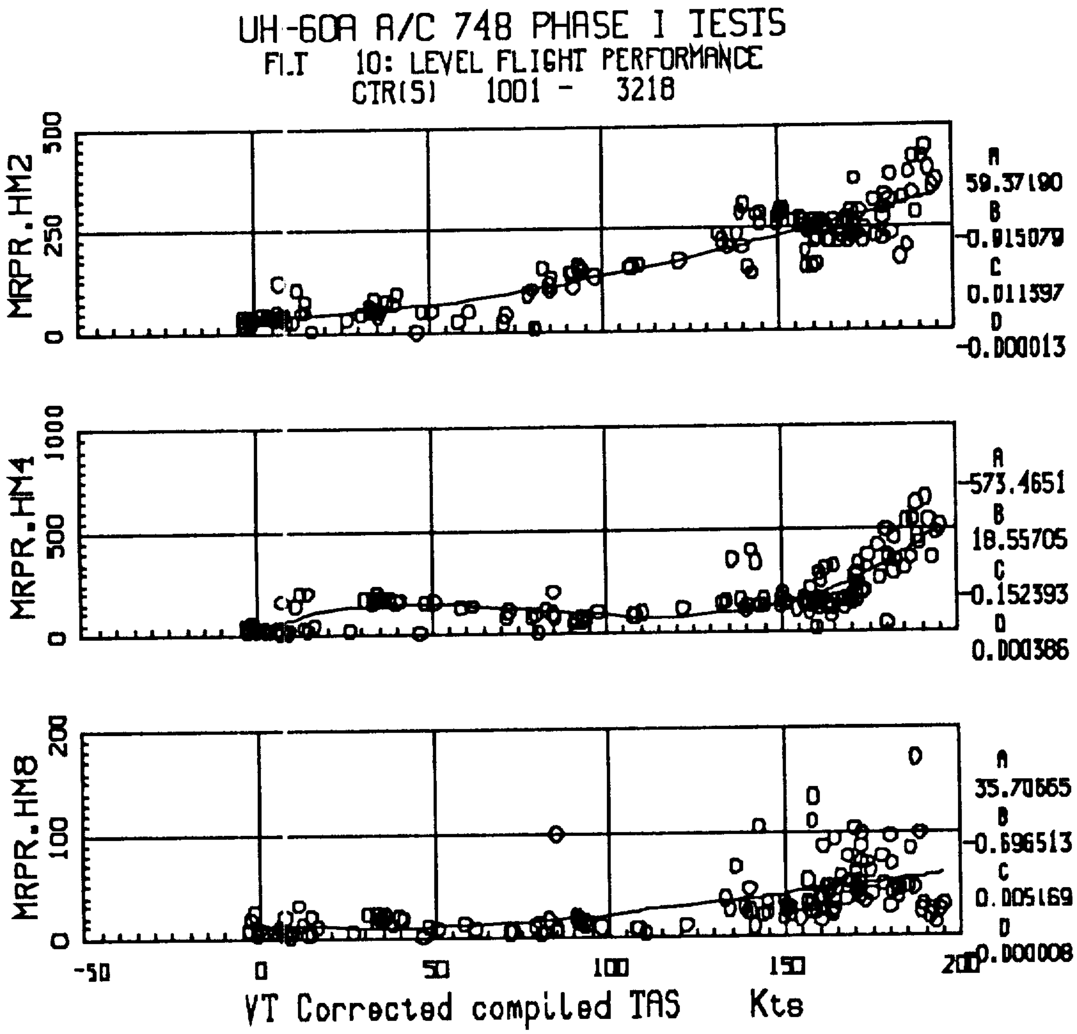


Figure 3.7. MULTIPLT Multifamily Plots



CTR 1004

Harmonic	Amplitude	Phase	MRPR
0	-454.3		
1	400.1	-51	
2	146.8	-21	
3	82.6	136	
4	66.6	-112	
5	20.8	- 21	
6	14.2	84	
7	13.4	- 5	
8	11.2	160	
9	4.4	97	
10	10.2	-57	
11	3.5	135	
12	4.7	-26	
13	7.3	-170	
14	12.6	76	
15	5.9	95	

Figure 3.8. HARMONIC Plot and Print Examples

3.2.6 Histogram Data Plots

TRENDS contains a capability for conducting some elementary loads analysis through the menu-item LOADS. This program feature uses a compressed data form called "minmax-per-rev" (MMR) data, sorted for certain vibrational data items to derive a frequency distribution. This frequency distribution is displayed as a printer-plot histogram showing the number of samples in each load range for the full range of oscillations measured over the counter. LOADS also shows the MMR data values plotted versus rev number for each counter. Plots produced by LOADS are shown in figure 3.9

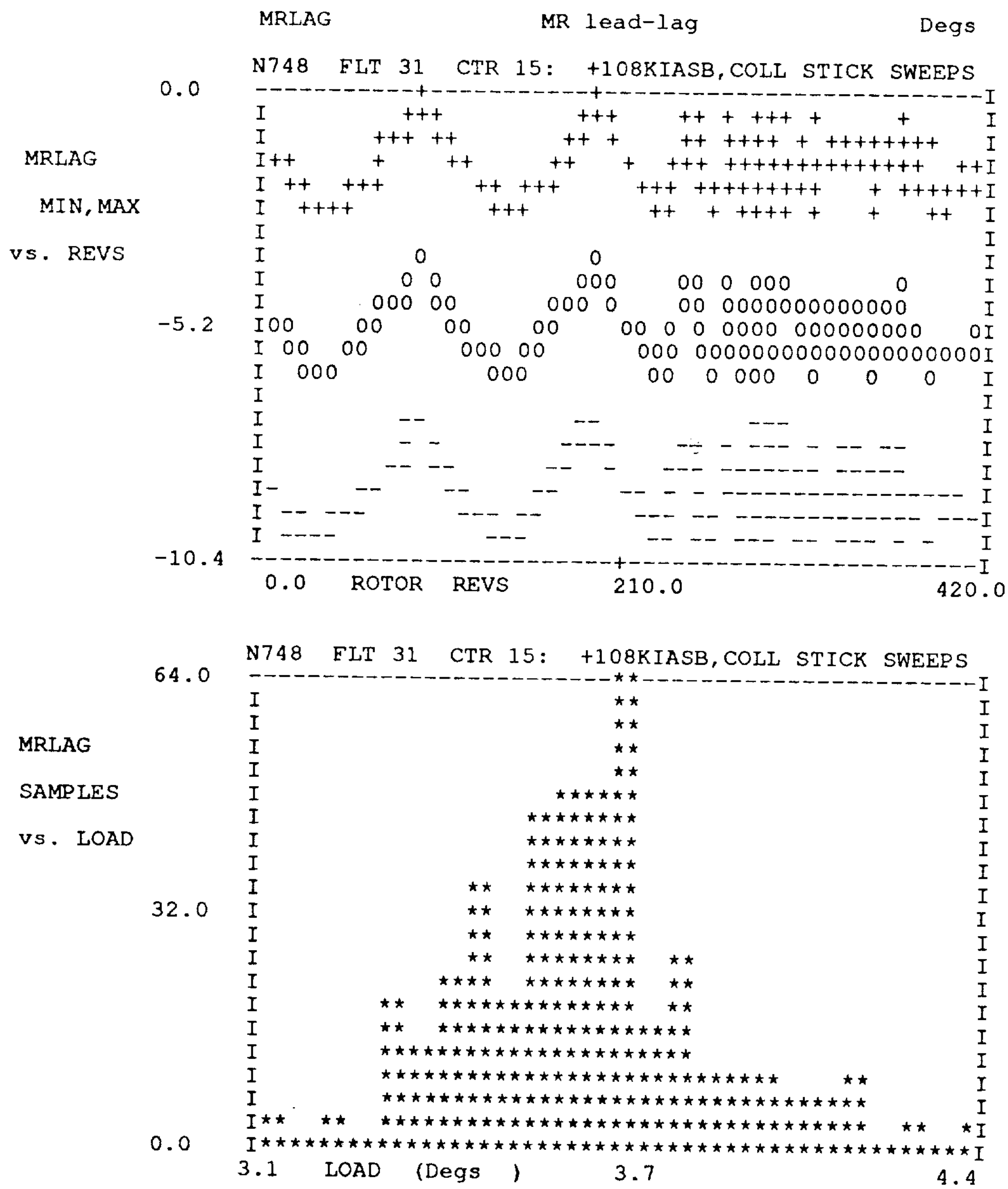


Figure 3.9 Load Plot Examples

3.3 DATA SEARCHING AND PSEUDO-FLIGHT GENERATION

TRENDS databases may contain thousands of test points or counters, representing many test conditions and test results. If the database were very small, users could just scan their notes or a listing to find what they wanted. With a large database, users must search the database for those conditions in which they are interested. TRENDS provides several ways of searching the database. A search on numerical data values (i.e., Min/Max statistics or harmonics) may be conducted with SEARCH. A search on text strings in the test-point descriptions is enabled through WORDSCAN. Flight descriptions may be searched using FLIGHTS or LOGSCAN. Availability of time-history data in the database may be determined through menu-item FIND. These search features produce a set of "success" counters when the search process finds the condition, text string, or available time-history data for which the user was looking. The set of "success" counters is often called a "pseudo-flight" or a "derived counter-set." Such a set of counters may be saved in the user's directory by name and then recalled to initiate another search (with different conditions), or a plot or tabulation. This search feature is one of TRENDS' most powerful and useful attributes.

3.3.1 Scanning Numerical Data

Numerical condition searches are carried out in TRENDS by SEARCH. This program feature lets the searcher define a "condition mask" or success template which defines a successful condition, then applies the condition mask to a specified data region (which may be the entire database, one or more flights, or a pseudo-flight) to locate counters which satisfy the desired conditions. The conditions are specified by defining allowable bounds (i.e., a success-range of values) for each test function. The test function is an expression in the Min/Max statistics or harmonic amplitudes (see 3.2.5). Of course, the test function may just be a simple statistic such as the mean value of airspeed. As many as 50 test functions are allowed for one search, but most searches use three or fewer. Condition masks may be saved by name for recall and (optionally) editing and application to another data region at a later date.

3.3.1.1 Example of SEARCH. The following is an example of a numerical search in TRENDS. In this example we will search the XV-15 (A/C 703) database for helicopter-mode test points for which the total average oscillatory mast torque exceeds 5% of the total average steady mast torque. Helicopter mode is signified by the pylon angle (D186) being above 85 degrees.

85 < D186 < no upper bound

The mast torque condition is

0 < (M143.OSC+M107.OSC)-(.05*ABS(M143+M107)) < no upper bound

The itemcode (e.g., D186) when written without an extension defaults to the average steady value. This example illustrates the use of functions of parameter statistics in defining the condition mask. Both of the conditions must be met for a successful search. A search initiated with this condition mask over flights 180 through 230 found 284 test points (counters) out of several thousand for which data were available for those flights in the A/C 703 database. This set of counters was then saved under the name OSCGT5PCT (along with a user-supplied description) for later recall as a data-region specification for plotting or further searching.

3.3.2 Scanning Narrative Data

TRENDS provides a capability for computer-aided scanning of flight and test-point descriptions as a means of locating data regions of interest for further exploitation.

Menu items FLIGHTS and LOGSCAN are used to scan flight descriptions. Flight descriptions include brief titles for each flight (e.g., "FERRY TO DRYDEN," "TRACK & BALANCE," "AEROELASTICS," "MANEUVERING FLIGHT") as well as structured ("PILOTS: DUGAN AND HARDY," "WEATHER: GUSTY") and unstructured ("NOTED OIL LEAK," "PILOTS LIKED THE NEW SIDEARM CONTROLLER") descriptive information. The following lines show some output from FLIGHTS.

```
AIRCRAFT: 703          H/Q AND PILOT TRAINING            T/O GW: 13656
  FLIGHT: 180 (G217)   LOCATION: ARC                         CG: 300.0
FLT DATE: 8 FEB 84     COUNTERS: 10254-10324        HRS TO INSP:   8.2
DIRECTOR: SCHROERS     PILOTS: GERDES/TUCKER/WILSON    FLT TIME:   2.0

FLIGHT INFO:    SCAS /HQ EVALUATION  -  PILOT TRAINING (WILSON)
                WIND: CALM     TEMP: 51' F     BARO: 30.06/-175

CONFIGURATION: NORMAL PREFLIGHT
               RH FUEL GAUGE CHECKED ON BENCH
               TAIL CAMERA INSTALLED

FLIGHT NOTES:  DIFFICULT TO TRIM WITH SCAS OFF
               PILOTS COULD NOT HOLD A/C IN 3/4"STEPS - 1/2" WERE OK
               A/C ROLLED TO THE LEFT IN LT&RT PEDAL STEPS
               A/C NOT HARD TO FLY WITH SCAS OFF - EASIER WITH SCAS ON
```

Users who are unfamiliar with the flight-test projects can use TRENDS' searching features to locate flights of interest; flight-test engineers at Ames who use TRENDS do not often use this searching feature because they remember the flights when certain key events took place. TRENDS does provide a depository for all of the flight-descriptive narrative data, however, so that project managers and other investigators have a readily available source of this information even when the flight-test engineer is unavailable.

The most useful descriptive-data search capability has proven to be the program feature called WORDSCAN, which searches the test-point descriptions for key words or groups of words such as "HOVER," "LEVEL," "STEP" or "SCAS." WORDSCAN shows the test-point's time-duration and the types of time-history data available in the database as well as the description. The test-point descriptions for the XV-15 database are taken from the brief notes on the pilot's card. They were not composed as structured strings with set keywords or codes to help in a computerized search, but they are very useful nevertheless for locating counters of interest because they are usually phrased or abbreviated in the flight-test vernacular. UH-60 test-point descriptions are somewhat better structured for computerized scanning. An example of WORDSCAN is given below.

Another type of narrative data which may be searched by the TRENDS user is the parameter-descriptive type. The menu-item called ITEMDEFS provides a capability for locating the itemcodes or mnemonics for the sensors of interest. A search for "ANGLE OF ATTACK," for example, would locate itemcode D008 for the XV-15 databases and mnemonic ALPHA for the UH-60 database. Users may also use ITEMDEFS to list all of the "CONTROL POSITION" itemcodes and their descriptions for XV-15 or "ROTOR PARAMETER" names for UH-60.

3.3.2.1 Example of WORDSCAN. This program feature enables the computer-aided search for particular words (i.e., character strings) in the databased test-point descriptions. Users may look for test points whose descriptions contain any of several specified words (an OR search) or all of several words (an AND search) or they may look for test points whose descriptions do not contain specified words (NEITHER and NOR searches). A search for "LEVEL" might miss descriptions for which "LEVEL" was abbreviated by "LVL," so the user might ask TRENDS to look for "LEV" or "LVL." As an example of the use of WORDSCAN, we will look for either "AFT," "FWD," or "F/A" in the test-point descriptions of flights 225 through 228 of the XV-15 (A/C 703) database. WORDSCAN prompts "Look for:" and we respond with "AFT,FWD,F/A." (Note the implied OR.) Then WORDSCAN prompts "Flights:" to which we respond "225-228." WORDSCAN finds 28 test points, some of which are

		Pilot Comments	Duration	T-H Data
Flt 228	CTR 13072	SLOW FWD ACCEL	52.849	HQ
Flt 228	CTR 13073	SLOW FWD DECEL	33.933	HQ
Flt 228	CTR 13090	1" AFT STEP	9.805	HQ
Flt 228	CTR 13091	1" AFT STEP	13.064	HQ
Flt 228	CTR 13092	1" FWD STEP	7.518	HQ
Flt 228	CTR 13093	1" FWD STEP	11.172	HQ
Flt 228	CTR 13094	2" AFT STEP	14.590	HQ
Flt 228	CTR 13096	1.5" AFT STEP	9.912	HQ
Flt 228	CTR 13097	1.5" FWD STEP	18.092	HQ
Flt 228	CTR 13098	3/4" AFT STEP SCAS OFF	12.032	HQ
Flt 228	CTR 13099	3/4" FWD STEP SCAS OFF	13.726	HQ
Flt 228	CTR 13108	3/4" AFT STEP	12.809	HQ

We save the counter numbers of these test points as pseudo-flight or derived counter-set FASTEPS. We then search this pseudo-flight for "SCAS OFF" and find five counters, which we save as FASTPSOFF for later control response investigations in TIMEHIST. (See the example in paragraph 3.3.3.1 below.)

3.3.3 Pseudo-Flight Generation Function

As seen from the preceding examples, the results of searches may be saved by name as pseudo-flights which may be used again to specify the set of counters on which to do further searches. Pseudo-flights derived from numerical-condition searches (SEARCH) are of exactly the same form as those derived from narrative searches (WORDSCAN). A pseudo-flight derived in SEARCH may be used to specify the counters to be used in searching for a particular text string in WORDSCAN and vice-versa. Each pseudo-flight is saved in the user's directory and is self-documented (i.e., the information is included along with the counters) by its creation date and a user-supplied description. Pseudo-flights may be concatenated or appended and stored under another name. A list of the user's stored pseudo-flights may be obtained using program-feature FILES. This program feature also enables deletion of pseudo-flights from the user's directory and copying of pseudo-flights from another user's directory.

3.3.3.1 Recall of Pseudo-Flights. The principal application of pseudo-flights is for plotting Min/Max statistics or harmonics (MINMAX, MULTIPLT, HARMONIC) for a set of related test points. TRENDS accepts pseudo-flights by name in response to the prompt for data region in any of the plotting, listing, or searching program features. For example, the prompt-response dialogue for data region in MINMAX might be

```
Enter flights: ALLHOVERS
```

where ALLHOVERS is the name of a set of counters whose test-point descriptions included the word "HOVER." Pseudo-flights may be used to specify the families of points for each curve in MULTIPLT. For example, one might use SEARCH to define pseudo-flights "PYLON30," "PYLON60" and "PYLON75" for families of test points flown at several different pylon angles, then use MULTIPLT to plot mast torque versus airspeed for those families.

Another application of pseudo-flights is as a series of counters for sequential time-history plotting applications. The data-region prompt-response dialogue in TIMEHIST might be

```
Enter counters: FASTPSOFF
```

where FASTPSOFF is the name of a pseudo-flight of longitudinal steps with SCAS off, derived by use of WORDSCAN. TIMEHIST would show the plots for each counter of the pseudo-flight one by one. If the "hardcopy-only" plotting option is selected, no user intervention is required to produce time-history plots for the entire pseudo-flight.

3.4 ANALYSIS

TRENDS could have been designed as a tool for simply storing, retrieving and displaying flight-test data, leaving the task of interfacing the data with analysis tools to the individual user. Some TRENDS users with specialized analysis tools prefer that mode of operation, but most appreciate having analysis tools integrated into the system. Some of the analytical capabilities of TRENDS are in-line functions (e.g., derivatives, filters, polynomial fits, formula evaluation), and others are expressed as gateways to other tools (e.g., DATAMAP, GTRSIM). Analytical capabilities have developed according to the needs of users and will probably increase in the future as other needs are recognized.

3.4.1 In-line Analysis Tools

Most of the in-line analysis tools provided with TRENDS are for operations on time-history data (e.g., derivatives, integrals, filters, spectral analysis, cycle averaging), but some (formula evaluation, polynomial regression) are applicable to either time-history or statistical (i.e., scalar) data. Each of these in-line tools is called out by the user via a structured syntax and/or keywords, with no requirement for recompiling the program or passing the data to another program.

3.4.1.1 Formula Evaluation. This extremely powerful and useful capability enables the plotting or searching of functions not explicitly stored in the database, but that are derivable from stored numerical data. The formulas may contain derivatives and integrals as well as library functions such as trigonometric functions, square roots, or table lookups. Users may simply want to rescale or bias a variable or change its units (to metric, for example) for plotting, or they may want to root-sum-square several parameters. These and many more possibilities exist within TRENDS under the category of formula evaluation. TRENDS contains a program feature (menu-item FUNCTION) for storing often-used or complicated formulas or expressions by name. These formulas may be used alone or within other in-line or stored formulas for plotting or searching applications.

3.4.1.2 Derivatives and Integrals. These calculus operations are among the operations permitted on time-series in TRENDS. These are useful for deriving rates or for integrating rates when the desired quantities have not been recorded in the test or for comparisons when both rates and position or angle data have been recorded. One use of the derivative has been to derive ground speed from recorded radar tracking position data during XV-15 acoustics tests. Integrals have been used to compare measured angular data with that obtained by integration of angular rate data. Implementation of these operations in TRENDS is as simple sums and differences of time-series, scaled appropriately by the time increment between samples. These operations may be used in conjunction with any of the other in-line operations with very few exceptions. One of the exceptions is that differentiation of an amplitude spectrum is not allowed, but an FFT of the derivative of a time history is legitimate. Derivatives and integrals are generally allowed in expressions and may operate on mathematical expressions as well.

3.4.1.3 Fast Fourier Transforms. The FFT is available as an analysis tool in SPECTRA and TIMEHIST for analyzing the frequency content of a given time-series. The FFT analysis may operate on formulas or expressions involving several time-series as well as on individual recorded parameter time-histories. The algorithm used in TRENDS was taken from DATAMAP.

3.4.1.4 Regression. Univariate polynomial regressions (i.e., polynomial fits) up to third order are available in both TIMEHIST and MINMAX. Regression is also automatically used to provide reference curves in HARMONIC. The polynomial least-squares fit uses the abscissa expression as its independent variable and fits the ordinate expression to it. The polynomial is evaluated for each abscissa point and line-plotted. The coefficients are not stored, but are displayed in the plot legends.

3.4.1.5 Convolution Filter. This digital low-pass filter may be used in TIMEHIST to remove unwanted high-frequency components from a time series. The convolution filter is not recursive, but operates with the entire time-series, using a window (selectable as either Hanning or half-cosine) to produce a result which exhibits no appreciable lag. The time-series upon which the filter operates may be an expression involving several time-series. The filtered result cannot be used in a formula or expression directly, but may be stored by name and counter for recall and potential use in formulas.

3.4.1.6 Cycle Averaging. Rotorcraft analysts are frequently interested in the behavior of some parameter as a function of rotor azimuth. TRENDS provides a capability (adapted from DATAMAP) for displaying rotor azimuth as the abscissa in TIMEHIST plots, using a technique known as cycle averaging. The averaging comes about because the time-histories differ somewhat from cycle to cycle (i.e., from one rotor rev to the next) and the analyst is usually more interested in the commonality than the differences, so the data from a specified number of cycles are averaged together for display. Each cycle is broken into equal increments of azimuth and then the measured data are interpolated to the times which correspond to each discrete azimuthal value. Neither the XV-15 nor the UH-60 instrumentation systems actually measures rotor azimuth, but rather provide a blipper or trigger (i.e., a bit in a code word) to signal when the rotor passes a certain point in its cycle. Rotor azimuth is derived from the blipper signal under the assumption that azimuth angle is a linear function of time over each cycle.

3.4.2 Simulation

Aircraft simulations are usually not integrated with flight-test analysis tools such as TRENDS and it is a major job to make other than gross comparisons of simulation and flight-test results. One non-real-time (i.e., off-line, strictly numerical) simulation, GTRSIM (Ref. 1), has been integrated with TRENDS. This "Generic Tilt-Rotor Simulation" is an Ames-developed derivative of a program originally developed at Bell Helicopter Textron (BHT). It models the XV-15 and could be made to model the V-22 and other tilt-rotorcraft as well. Future effort will be directed toward integrating other simulations with TRENDS.

3.4.2.1 GTRSIM. This simulation is capable of finding trim conditions and displaying control responses (among other things) as functions of specified model parameters, control inputs, and flight conditions for the XV-15. An interface has been made with TRENDS so that the analyst can conveniently access flight conditions and control-position time-histories from the XV-15 database to use as inputs to GTRSIM. The analyst may then run GTRSIM and store the results for comparison with flight-test results. A correspondence has been made between simulation-variable mnemonics and flight-test itemcodes, including differences in units between the two data sources.

3.4.3 DATAMAP

DATAMAP, "Data from Aeromechanics Test and Analytics--Management and Analysis Package," (Ref. 2) is a powerful and elegant tool for analyzing rotorcraft test time-history data. Originally developed for the Army by BHT, DATAMAP has been modified and improved at Ames. Some of its capabilities are duplicated in TRENDS (e.g., spectral analysis, convolution filtering, cycle averaging), but it has many more analytical capabilities. An interface has been provided between TRENDS and DATAMAP to enable DATAMAP to access data from TRENDS databases so that serious analysts can use TRENDS to locate data regions of interest, then move to DATAMAP for detailed investigations.

3.4.3.1 Capabilities of DATAMAP. A brief list of DATAMAP's features follows.

ANALYSIS OPTIONS

Amplitude Spectra	Stochastic Process Analyses
Harmonic Analysis	Frequency Response Function
Digital Filtering	Coherence Function
Moving-Block Damping	Auto-spectral Density
Cycle Averaging	Auto-correlation
Min/Max Analysis	Cross-correlation
Acoustic Analyses	Cross-spectral Density
Perceived Noise-Level	Basic Statistical Analyses
Narrow-band Analysis	Mean
Octave Analysis	Variance
Third-Octave Analysis	Standard Deviation
A,B,C,D Network Integration	Chi-square Test for Normal

DERIVED PARAMETERS

True Airspeed	Blade Static Pressure Coefficient
Rotor Azimuth	Blade Normal Force Coefficient
Rotor RPM	Blade Chordwise Force Coefficient
Shaft Horsepower	Blade Pitching Moment Coefficient
Shaft Thrust Coefficient	Blade Displacement
Shaft Torque Coefficient	Blade Local Flow Magnitude
Density Altitude	Blade Local Flow Direction

GRAPHICAL FEATURES

X-Y Plots:	Multiple curves on one plot (see figure 3.10)
	Linear or log scales in either direction
	Small data windows can be selected
	Tektronix cursor can be accessed to measure points
Contour Plots:	Rectangular or cylindrical format
	Interval, initial level, density are all selectable
Surface Plots:	Rectangular or cylindrical format
	Surface may be viewed from any angle
General:	Conversion of dependent variable units

DATAMAP PLOT

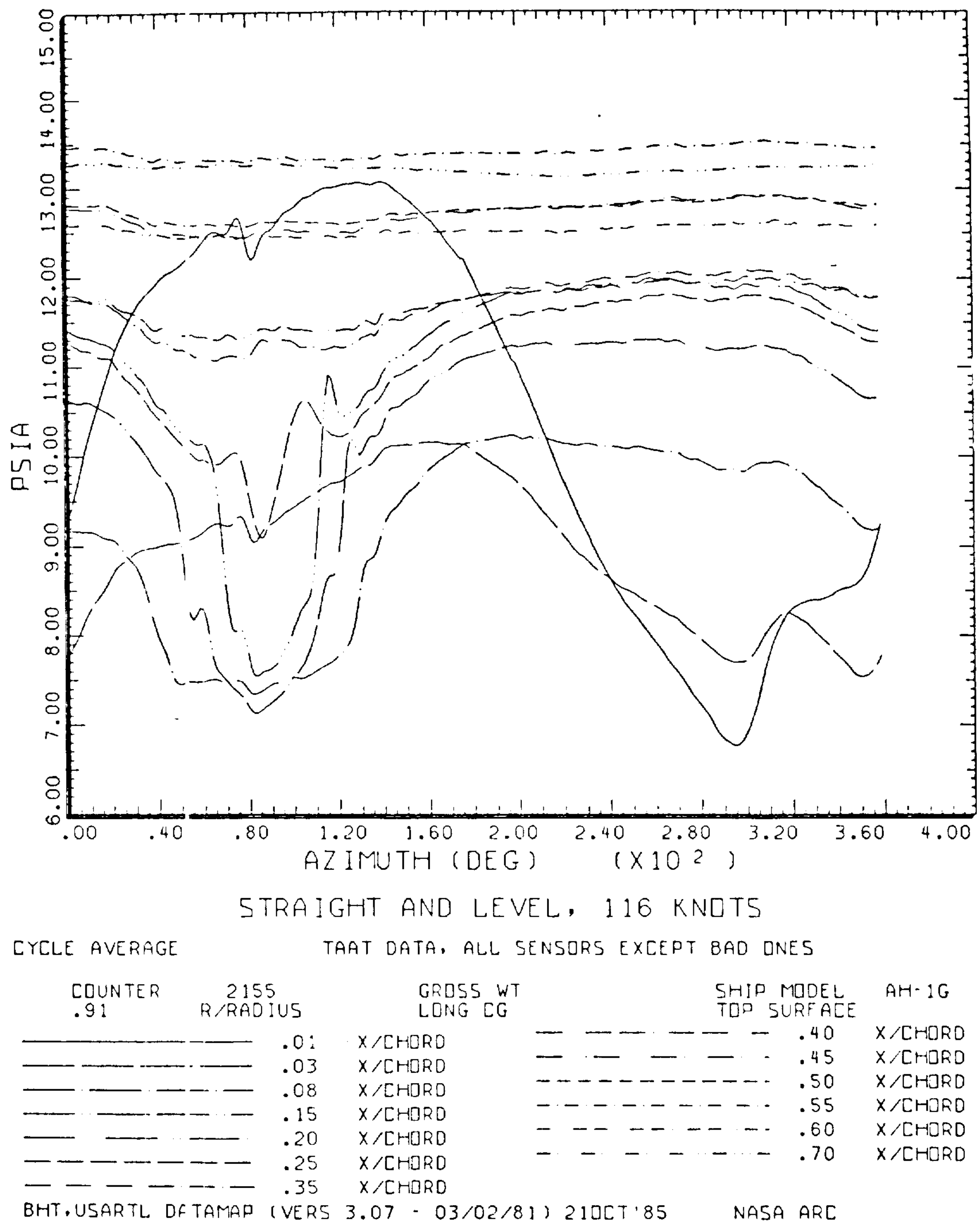


Figure 3.10 Plot Examples from DATAMAP.

Section IV

DATABASE MANAGEMENT

4.0 DATABASE MANAGEMENT INTRODUCTION

This section deals with acquiring, storing and maintaining flight data in the TRENDS database. The TRENDS philosophy is that the Database Manager is in charge of these tasks, so that the flight test engineer, analyst, or project manager is free of these responsibilities.

In our system the Database Manager is tasked to gather the preprocessed digital tapes from the flight-test center; input the correct parameters per test point for filtering and decimation; enter and reformat the data into the TRENDS format; compute the derived parameter quantities and check the data validity. These steps are semiautomatically performed by using the database management menu; once the parameter sets and counters have been decided upon, most of the processing is automatic. Validity checks are automatic, but if bad data shows up, the Database Manager must look at the data plots individually. The connection between the database management software and the accessing software of TRENDS is illustrated in figure 4.1. Descriptions of menus, menu items and solutions to problems involving database management will be addressed in this section.

4.1 DATABASE MANAGEMENT MENUS

Database management menus, similar to the data-accessing menu of TRENDS have been developed to integrate the database management tasks. Separate menus exist for the XV-15 and UH-60 database management functions because the procedures differ significantly. Future efforts will be undertaken to investigate commonalities in the processes and to develop an integrated database management menu.

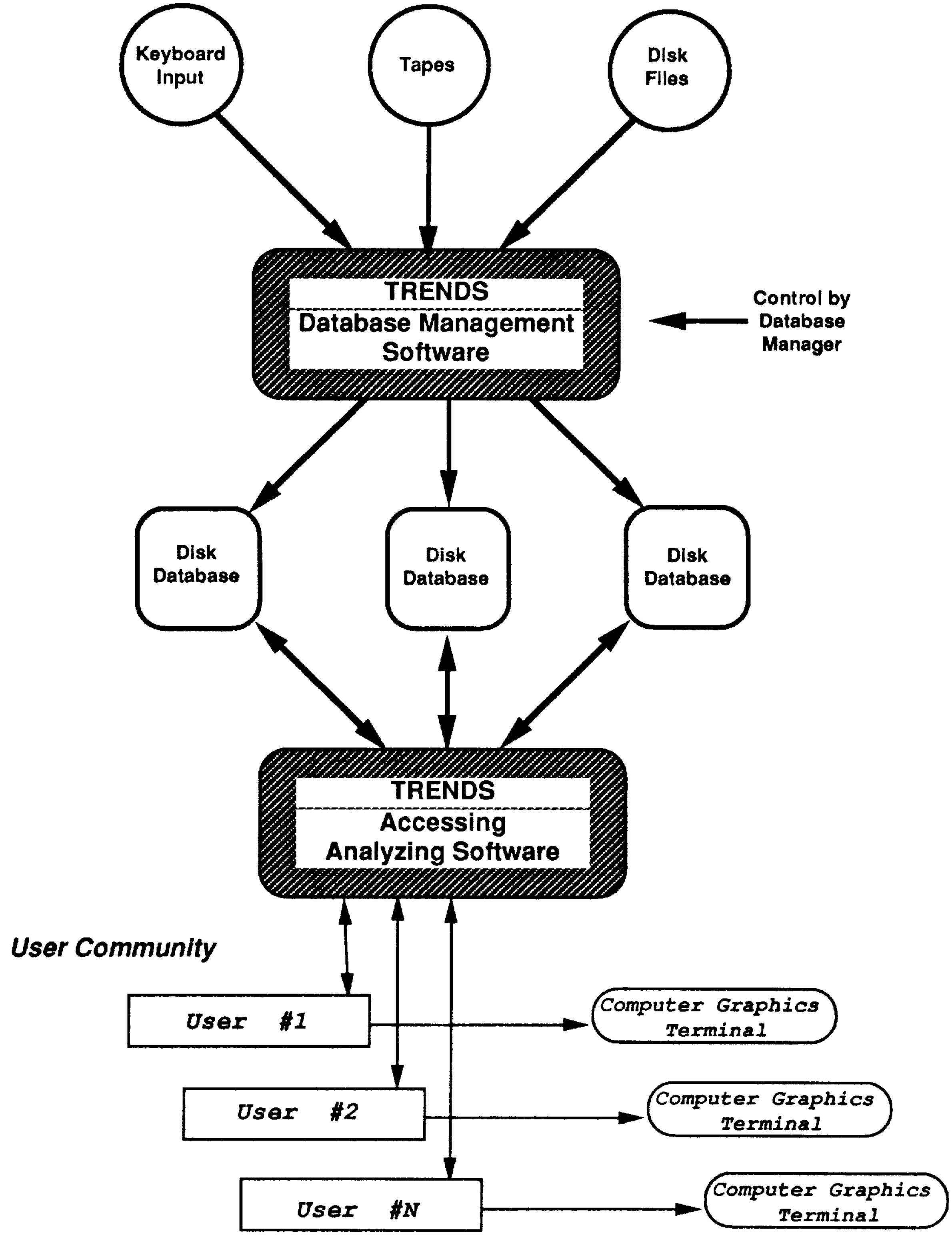


Figure 4.1 Relationship Between Database Managing Software and Accessing Software

4.1.1 XV-15 Database Management Menu

The database management menu for the XV-15 database is

Flight Data Input	Display Files	Summarize	Data Handling
FLIGHT DESCRIPTIONS	CHECK T-H	FSTATUS	COPY
TEST-POINT NARRATIVE	VIEW MMC	CONSISTENCY	REPAIR
DATA REQUEST	KEY FILE	SUMMARIZE	DELETE
MINMAX DATA	CTRCAT	BATCH CHECK	INPUT LIST
DERIVED DATA	THFCAT	REPORT	TAPE LIST
GROSS WEIGHTS	ICS BY CTR	VALIDITY CHECKS	DESPIKE
TIME-HISTORY DATA	WORDSCAN	SCREEN	FILTER
SET DIRECTORY	ITEMDEFS	TRENDS	GROUPS
HELP	SEARCH	TMPTRENDS	NOMINALS
EXIT			

4.1.1.1 Menu Items. The functions of the menu items are

Keyword	Description of Keyword
Flight Data Input	
FLIGHT DESCRIPTIONS	Enter descriptions for each flight
TEST-POINT NARRATIVE	Enter a short description for each test-point
DATA REQUEST	Display or enter requests for T-H data by flight
MINMAX DATA	Reformat the Min/Max list file produced on FOX VAX
DERIVED DATA	Calculate and store additional Min/Max data
GROSS WEIGHTS	Enter initial gross weight and center of gravity
TIME-HISTORY DATA	Process data from FOX-produced compressor tapes
SET DIRECTORY	Set aircraft tail number; change file locations
HELP	Display purpose of menu items
EXIT	Return to VMS
Display File Contents	
CHECK T-H	Display contents of individual T-H files--one line/ctr
VIEW MMC	Display contents of Min/Max files--one line/counter
KEY FILE	Display contents of FLTCTRxxx.KEY files
CTRCAT	Display contents of COUNTER.CAT--one line/counter
THFCAT	Display contents of THFILES.CAT
ICS BY CTR	Display contents of COUNTERS.xxx files
WORDSCAN	Call WORDSCAN to search/display counter descriptions
ITEMDEFS	Call GROUPITEM to display/search item definitions
SEARCH	Search values in Min/Max files

Summarize

FSTATUS	Display by flight total counters by data group Tally and display counters per group for all flights
CONSISTENCY	Display status of data and support files by flight
SUMMARIZE	Summarize the available data in the database for a given flight
BATCH CHECK	Create and submit command file to check database status
REPORT	Produce a printable report file Include summary of contents and storage of the database
VALIDITY CHECKS	Compile and process "Summary of Data Spikes" Pinpoint anomalies of data values for further analysis
SCREEN	Test the value of data points (from Min/Max files) against a standard list of boundary values (NOMINALS)
TRENDS	Call TRENDS and leave this utility, pointing to data in the XV-15 TRENDS database
TMPTRENDS	Call TRENDS and leave this utility, pointing to processing location of the data

Data Handling

COPY	Copy T-H counters from elsewhere by flight or counter
REPAIR	Correct or alter values for data in the database
DELETE	Delete T-H flights or counters from the database
INPUT LIST	Display/edit itemcode lists + filter freq. + decimation Read and modify files used by the DCMPRS program
TAPE LIST	Display/edit list of tapes with CCF Library VSNs
DESPIKE	Perform search for spikes on raw data files Replace data values and write to another file
FILTER	Use the convolution filter on specified files of data
GROUPS	Display/search items by groups
NOMINALS	Display, search, or modify a standard list of normal limits of valid data values for selected itemcodes

4.1.2 UH-60 Database Management Menu

The database management menu for the UH-60 database is

Flight Data Input	Control/Edit	Database	Utilities
FLIGHT DESCRIPTIONS	HOT LIST	FSTATUS	SCAN
TEST-POINT NARRATIVE	DEFINE ITEMS	ITEMDEFS	COPY
NUMERICAL DATA	HELP	WORDSCAN	DELETE
TAPE-READ	COMPRESS	TRENDS	MASK
LIST MES,EU FILES	EXIT	INSPECT	

4.1.2.1 Menu Items. The purpose of each of these menu items is

FLIGHT DESCRIPTIONS	Enter flight description data (BASKER)
TEST-POINT NARRATIVE	Enter test-point descriptions (INPCOM)
NUMERICAL DATA	Run FILLER to reformat AEFA data for TRENDS
TAPE-READ	Tape-to-disk commands to load AEFA data on NEP
LIST MES,EU FILES	Display contents of MES748fff.DAT and EUfff.DAT
HOT LIST	Enter items, filter, decimation for T-H storage
DEFINE ITEMS	Enter item-descriptive data (FILLDINFO)
HELP	Display purpose of menu items
COMPRESS	Compress the ITH files
EXIT	Return to VMS
FSTATUS	Display status of support files in the database
ITEMDEFS	Call ITEMDEFS to display/search item definitions
WORDSCAN	Call WORDSCAN to search/display counter descriptions
TRENDS	Call TRENDS and leave this utility
INSPECT	Display contents of specified *.TIM files
SCAN	Display T-H counters stored for a flight or counter
COPY	Copy T-H counters elsewhere for a flight or counter
DELETE	Delete T-H flights or counters from the database
MASK	Comment out specified T-H flights or counters

4.2 INSTALLING DATA INTO TRENDS

TRENDS databases are made up of selected groups of parameters on the basis of each test point. Decisions have to be made by the Flight Test Director or his assistants as to which data (of those available) are to be included in the database. It is during this selection process that the first major data compression is performed. The quality of the data has to be monitored and steps have to be taken to correct problems where possible and exclude bad or misleading information from the database. Derived data to be accessed from TRENDS (e.g., Min/Max statistics, nonrecorded-but-derivable standard measures, harmonics) must be computed and stored. All of these tasks fall under the function of installing data into the database.

4.2.1 Formats and Media

The media on which data come to TRENDS include magnetic tapes, disk files, and printouts (or typed or handwritten pages). The most common source is magnetic tape. Tapes come in several formats, among which the following have been used:

1. Compressor tapes from ARC (tagged integers)
2. Compressor tapes from AEFA (tagged floating point)
3. RTM 45 tapes from Crows Landing (positional integers)
4. Bell Standard-label (positional integers)
   a. time-histories
   b. harmonic time-histories
   c. Min/Max-per-rev
5. Ames/Bell flight logs (ASCII descriptions).

Some of the data arrive as disk files:

1. Engineering Data Files (EDFs) (positional floating point)
2. Min/Max listing files from OFL (formatted floating point)
3. Wind tunnel data for HARP and BV360 (formatted floating point)

4.2.1.1 Reformatting Programs. A generic program for reformatting has not yet been implemented for all data formats, although this is desirable. In addition to reformatting the time-history data, reformatting programs perform other auxiliary tasks, such as deriving unmeasured parameters, filtering, decimating, computing Min/Max statistics, etc. Many of the processes are duplicated for each reformatting program, which in hindsight is a poor practice because of the upkeep required for one program versus that for many, but the programmer is sometimes seduced into nonoptimal choices when trying to develop software rapidly.

4.2.1.2 Processing Time. Compressor-tape processing time, even on a VAX 8650, is significant. The following computation is a guide for the XV-15:

CPU time(sec) = (10 + 1.5 x items) x counters

or

1 hour of CPU time ---> for 90 parameters and 25 counters

where the counters are from 15-20 sec long, the average sample rate is 100 samples/sec and where "items" represents the number of parameters selected for storage as time-histories.

4.2.1.3 Prompting Programs. Some portions of a TRENDS database (in particular, the narrative portions) are entered from printed sheets by someone who types the data in at a keyboard. The data are stored in TRENDS in binary, keyed-access files which cannot be changed with the system editor. Prompting programs (in particular, BASKER for detailed flight data and INPCOM for test-point descriptions) are provided via database management menus to assist the data-entry person in the task of filling each field in the proper format. Narrative information is important in the TRENDS database; hence, good tools for data entry increase the likelihood that the narrative information will be entered properly.

4.2.2 Data Storage Capacity Considerations

An important practice that not only improves data compression, but is implicit in the rapid processing of flight data, is the use of the prime-data bit and counter reference point in aircraft data acquisition systems. Implemented in the cockpit of each XV-15 aircraft is a push-button which the pilot presses when he is ready to take his test-point data. Upon pushing the button, a bit (the "prime-data" bit) is set high in the data stream. When the button is pressed again the prime-data bit drops low and a test point counter (CNTR), which is a separate digital word in the data stream, is incremented. Hence, onboard the aircraft in real time both the prime test period and counter numbers are automatically recorded in the data stream. The counter number thus identifies the prime-data time interval of interest. This interval, called a burst, run, maneuver, or test point by different people, is usually called a counter in TRENDS vernacular. The flight data, including the prime-data bit and counter number, are telemetered down to the ground test facility and output to stripchart time-tick per recorders, where test engineers monitor the test's progress. If the test director feels that a test-point is unsatisfactory, he can ask that it be repeated, identifying it by the counter number. The counter number and the prime-data bit are both used in post-flight processing to locate intervals of interest in the data stream. When the prime data bit and counter number are available, time-of-day bounds do not have to be specified to identify the data interval of interest. A list of bad test points to be skipped during post-flight processing is given to the ground data center immediately after each flight. TRENDS uses the counter number to key all of its test-point data, thereby relating test-point narrative to all types of numerical data.

Not all of the flight-test data from a large project can be stored for access by TRENDS users. Bad counters and non-prime data have not been included, for example. Because of the high data-acquisition rates (up to 500 samples/sec for the XV-15 and 2000 samples/sec for upcoming UH-60 tests), the additional measures discussed below are taken to further reduce data-storage requirements.

4.2.2.1 Counter Selection. Not all of the counters (test points) which are recorded in tests are included in the databases. If problems arise in instrumentation, data reduction, or test procedures to negate the value of a test point, or if a test point is judged to have no value relative to test objectives, it will not be included in the database. Decisions about what is to be included are made by flight-test engineers who are familiar with the test objectives and test results. For the XV-15 case, these engineers are provided with a list of the counters (with descriptions) as a file on the VAX. They edit the file to form a request specifying which counters and time-history groups are to be included in the base. Database management personnel then fill the database according to the request.

4.2.2.2 Groups. One means of reducing storage requirements is by definition and selective storage of time-history groups. Of the 350-odd parameters usually recorded for XV-15 tests at Ames, only about 90 are of general interest as accessible time-histories in TRENDS. These are the "performance" or "handling qualities" parameters such as airspeed, stick positions, etc. For some test points, not even these need be stored as time-histories because the Min/Max statistics provide enough information to satisfy the test objectives. For other test points, different groups of parameters are important to save, depending on the particular objectives associated with the flight and test point. The research engineers were consulted to determine which groups should be established, the data items making up each group, and appropriate filter cutoff frequencies and sampling (decimation) rates.

The following groupings have been defined for the XV-15 database:

ITEM CODE GROUPS	time-histORY FILE DESCRIPTION
AEROELASTIC	FILTERED, FULL CNTR, 26 items
CONVERSION	UNFILTERED, FULL CNTR, 23 items
HANDLING QUALITIES	FILTERED, FULL CNTR, 70 items
HARMONICS	UNFILTERED, 1000 PTS/CNTR, ABFMS
MANEUVERS	FILTERED, FULL CNTR, 88 items
SPECIAL AEROELASTIC	UNFILTERED, FULL CNTR, 10 items
SPECTRALS	UNFILTERED, 1000 PTS/CNTR, ABFMS
TRANSFER FUNCTION	UNFILTERED, FULL CNTR, 13 items

The XV-15 groups are defined in greater detail in appendix C.

A new concept, the "HOT-LIST" concept, was implemented in February 1987 to enable the flight-test director to change the parameters to be stored, their filter cutoff frequencies, and their sampling rates on a flight-by-flight or counter-by-counter basis as the database is filled. This concept has been implemented only for use with the UH-60 database.

4.2.2.3 Integer Storage. One means of data compression used by TRENDS is to store all time-history data as scaled 2-byte (16-bit) integers as opposed to 4-byte floating-point engineering-unit values. Most of the data are acquired with only 12 bits (or fewer) of precision anyway, so no information is lost by limiting the stored precision to 16 bits. Engineering-unit scaling is applied when the information is retrieved for use in TRENDS. Scales and biases are stored along with the integer data. This storage technique essentially doubles the storage capacity of the given disk space over that of floating-point storage. The trade-off is a small penalty in computer time to float and scale the data. When the input source is in calibrated floating-point form, as it is with UH-60 data from AEFA, TRENDS determines the range of data for each sensor and counter, then derives scales and biases which will serve to rescale the data when they are stored as 2-byte integers.

4.2.2.4 Filtering and Decimation. Another method of data compression used by TRENDS is the filtering and decimation of some time-histories before they are stored in the database. The convolution filter is applied to specified parameter time-histories, then only every nth sample of the filtered time-history is stored. The determination of needed bandwidth and sampling rate is made by knowledgeable engineers. The lower plot in figure 4.2 shows the filtered representation of the raw data shown in the upper plot. The effective time interval between samples is stored as part of the data. Synchronization of the data for two or more time-series being cross-plotted or used together in formulas is accomplished in TIMEHIST by interpolation of less-frequently sampled series to the sample times of more-frequently sampled data.

4.2.2.5 Series Truncation. Still another means of compression used by TRENDS is the truncation of certain time-series before they are stored in the database. The primary use of many recorded parameters (e.g., acceleration or bending moment) is to measure their frequency content. The analyst wants to know this information relative to an interval of steady flight at some particular flight condition, so it is useless to store the data across a transient part of the test point. A short burst of data is all the analyst wants, so it is both economical and reasonable to truncate the data to a short time interval. In the case of UH-60 data, a "cut-and-save" process is used to extract a short interval of the most useful data from the longer recorded test point for which data are available. This interval is determined by looking at TRENDS plots of certain low-frequency parameters temporarily stored for the full counter.

4.2.2.6 Rotor Rev Averaging. Pressure data from upcoming UH-60 tests will be recorded at 2000 samples per second, generating big storage problems. It is anticipated that these data will be compressed by averaging three or more rotor revs together to generate one averaged rotor rev of pressure data, hence significantly compressing the data to be stored.

4.2.3 Data Quality

There are always problems with real test data. Instrumentation is imperfect. Measurements are imperfect. Time-code data often contain anomalies which affect the processing of all test data. Data reduction is prone to errors. Good data differ from bad data only in the degree of imperfection. To avoid including "bad" data in the database, either the computer or a knowledgeable engineer must make a judgement. The problems of database management concerning the quality of test data are

1. How can bad data be detected?
2. How can bad time code data be detected?
3. Once detected, how can bad data be corrected?

4.2.3.1 Checking Bad Data. Given the time, tools, and interest, a knowledgeable aeronautical engineer can look through flight-test results and locate most of the problems with the data. We would like to have computer software which could do the same thing, but we have not yet succeeded. Users are encouraged to report suspicious data to the Database Manager so that it can be corrected, flagged, or deleted. Users are also reminded that they are looking at real test data in TRENDS.

Certain gross checks are made automatically by the database management software of TRENDS. For example, the XV-15 pylon angle is limited to the range of about 0 to about 95 degrees, so software can flag a problem if this parameter measures, say, 234 degrees. If the problem turns out to be a bad bias value, this can be corrected. Such checks are not generic, but are application-specific. Another check made by TRENDS is to monitor the change between time-history samples (a slope check) and report instances when such changes exceed preset bounds. When the bounds are exceeded, raw data are stored (even if storage of raw data had not been requested) for later plotting and investigation. Band-edge problems can also be detected and flagged automatically. Problems which persist are brought to the attention of the flight-test and instrumentation engineers.

The band-edge problem shown in the upper plot of figure 4.2 was caused by an improper transducer bias on the pilot's g-load signal (XV-15 itemcode A019). The problem was detected during automatic data processing and reported to instrumentation personnel for correction. The lower plot shows the filtered version of the data in the upper plot. This version, derived using the convolution filter and a cut-off frequency of 2 Hz, does not exhibit the band-edge problem.

4.2.3.2 Despiking Data. TRENDS' database management software includes provision for despiking time-histories and then for recomputing derived statistics and filtered-decimated time-histories from the despiked data. Single-point spikes (i.e., wild points, outliers) and some other errors which can be detected can be replaced by interpolating between the wild point's neighbors. Figure 4.3 illustrates a time-series with spikes (the upper plot, autoscaled) and the same time-series with the spikes removed automatically.

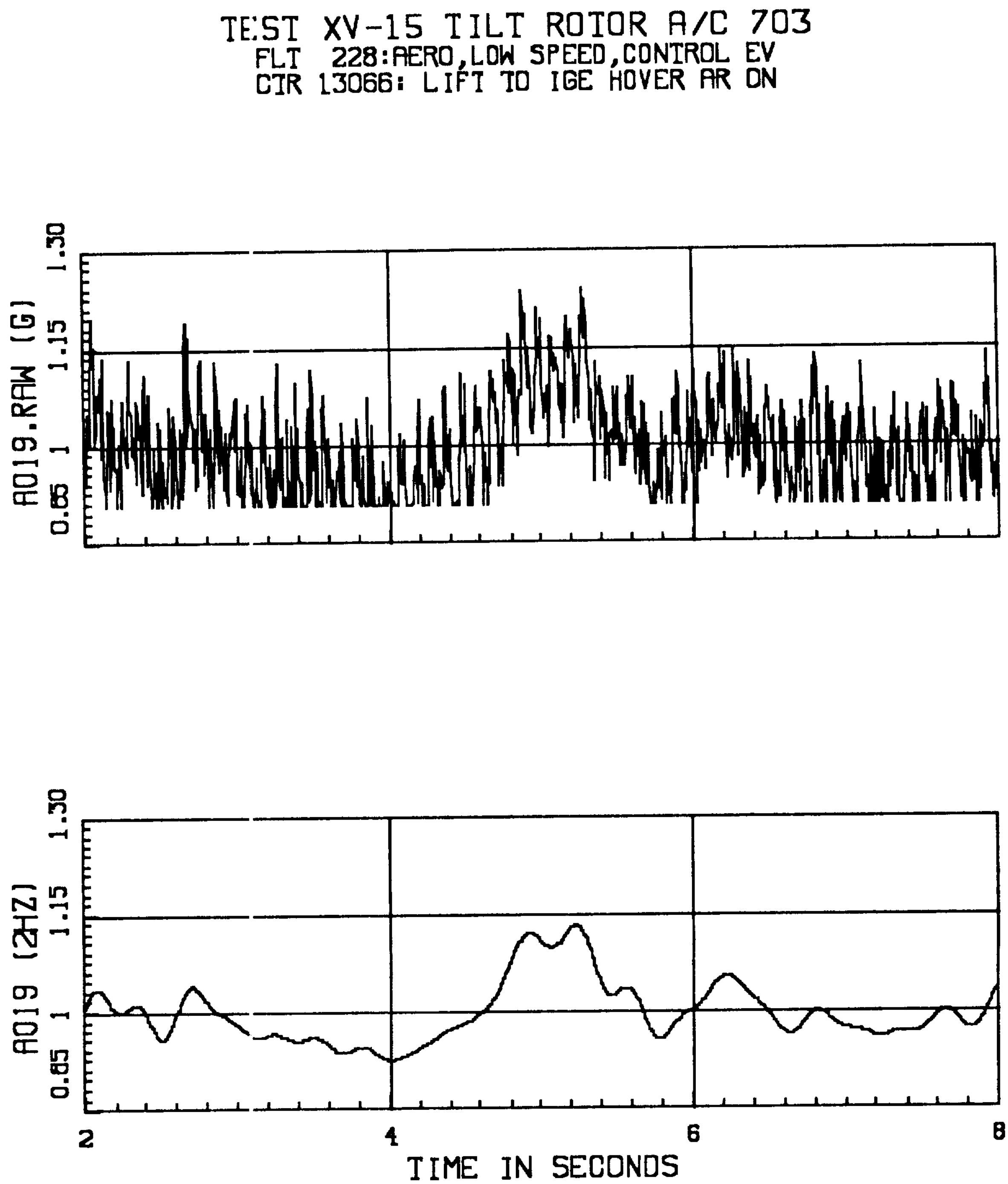


Figure 4.2 Band-Edging and Filtering Example

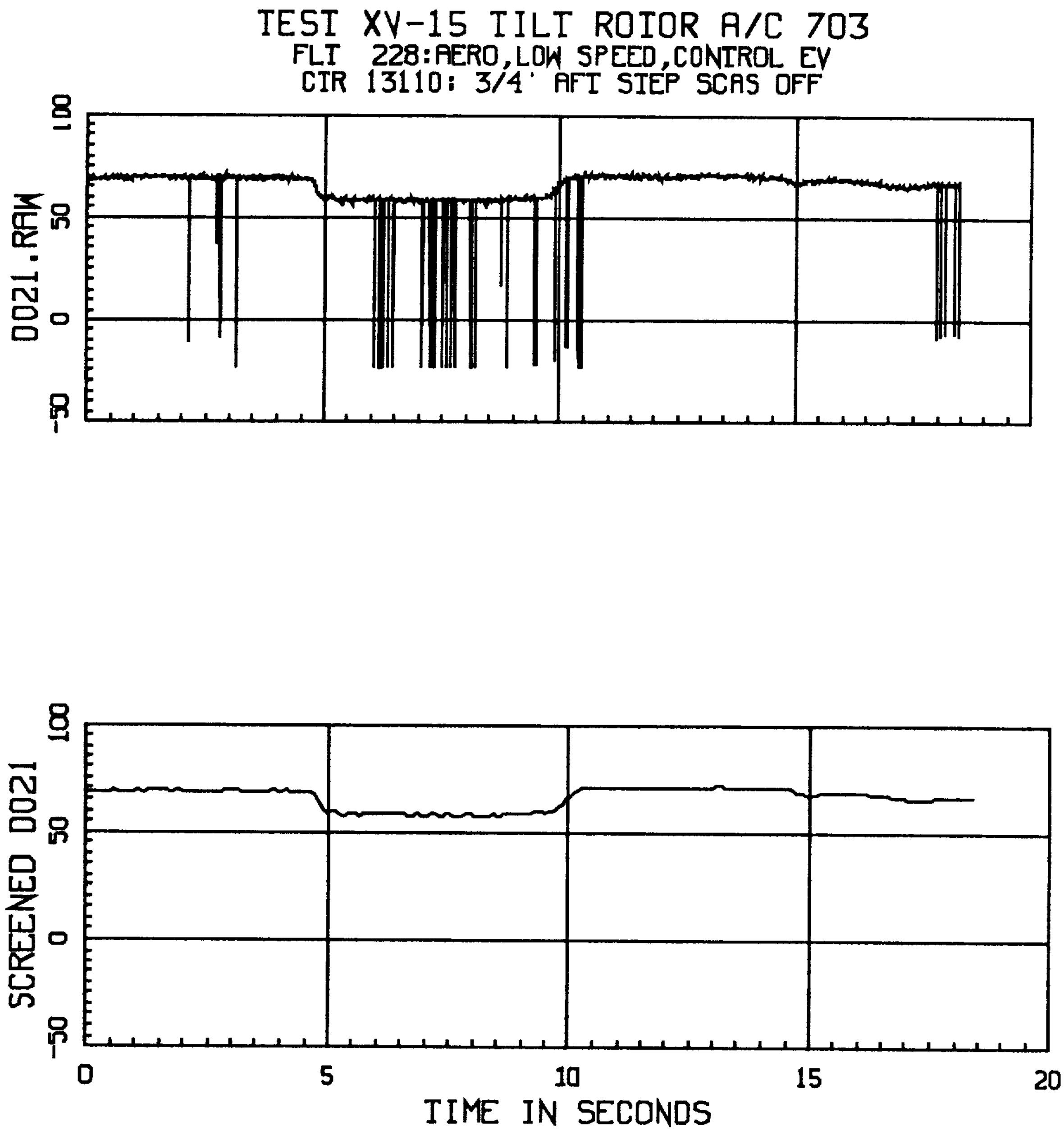


Figure 4-3. Example of Despike Program

4.2.3.3 Data Credibility. In the early days of TRENDS' development, users and potential users were not confident that the database would give them correct answers. Those XV-15 engineers had some degree of confidence (justified or not) in the printed Min/Max statistical reports provided to them by the Ames Ground Data Center (Code OFI), because the reports compared adequately with stripcharts which were generated during the flight. The statistical reports were ASCII files from which data could be extracted for inclusion in the database and the reports were usually available before time-history tapes could be obtained and processed to derive the Min/Max statistics. These ASCII files were selected as the source of Min/Max statistical data for TRENDS primarily because of their availability. However, a side benefit of their use was soon realized by the TRENDS managers when we saw the user community confidence in TRENDS rapidly increase because the TRENDS values were indeed the same as in the original printouts.

4.2.3.4 Time Code Problems. Although the aircraft time code generator is not a prime consideration in the way that TRENDS stores and accesses data, it cannot be neglected. In our data-acquisition system at Ames, that part of the time code which should logically contain days contains the counter number instead. Although it is less likely to happen than a glitch in the seconds word (which changes more rapidly), an incorrect counter sequence number caused by a time anomaly affects the unpacking of all data. This can cause data to be stored in the wrong location and it can wreak havoc with any automatic-processing logic. We have no general solution to this problem, but it is mentioned here because a significant effort can be required to solve such problems as they come along. It is useful to have counter number imbedded in the time code, because time code is telemetered to the ground and the counter number can then be used to identify data regions on the strip charts. It may be noted that data are stored not on the basis of time, but on the basis of sample rate in the data-acquisition system, since the rate is more reliable. Parameter files in TRENDS define what the sample rate is for each parameter (e.g. 251 samples/sec), so samples are separated in time by the reciprocal of the sample rate. Time is not stored per se in TRENDS files, except for the time of the first sample of each parameter. This initial time may differ between parameters (i.e., time skew), so it is recorded to enable the synchronization needed for cross-plotting or parameter combination in formulas.

4.2.3.5 Check Programs. As the size of a database increases, it becomes increasingly important to have tools which will automatically perform all of the checks which can be made automatically. Inconsistencies creep into the database unavoidably through software oversights, source-data problems, etc. It is necessary to have software which will locate inconsistencies and other problems so that they may be corrected. In particular, summary files must truly summarize, must be complete, and must not indicate data which do not exist. Several check programs (e.g., CONSISTNC) are used in managing TRENDS databases. These are documented in the TRENDS Procedures Manual, available from either of the authors of this report.

4.2.4 Data Derivations

A TRENDS database includes scalar numerical data of several types as well as time-history data. These types include Min/Max statistics, parameters derived from those statistics, parameters derived from time-histories (e.g., slopes), and harmonic amplitudes and phases. To derive these in line as the user runs TRENDS is computationally inefficient and impossible if the needed time-histories are not stored. In addition to the scalars, some time histories are derived for UH-60 storage and Min/Max-per-rev data are derived for loads parameters.

4.2.4.1 Min/Max Statistics. The Min/Max statistical measures for the XV-15 and UH-60 databases were listed in paragraph 3.2.5. The values for these measures are obtained by processing time-history data, using algorithms which are generally specific to the project and database. Sometimes the Min/Max source is an ASCII file of preprocessed values rather than a time history.

4.2.4.2 Derived Parameters. The derived parameters for the XV-15 are

Itemcode	Description	Units
AILD	RT AILERON RATE (SLOPE D645)	DEG/S
ALFD	ANGLE OF ATTACK (D008) RATE	DEG/S
BETD	SIDESLIP RATE (SLOPE D007)	DEG/S
CDUR	COUNTER DURATION	SECOND
CPXX	POWER COEFFICIENT	
CRPM	COMPUTED RPM	RPM
CTXX	THRUST COEFFICIENT	
DNLD	DOWNLOAD COEFFICIENT	
ETIM	ELAPSED TIME SINCE ENGINES ON	MINUTE
FLPD	FLAP ANGLE RATE (SLOPE D617)	DEG/S
GOVD	GOV. LVDT RATE (SLOPE E719)	%/S
GWJW	GROSS WT USING ADJ FUEL WEIGHT	LBS
GWT0	RAMP GROSS WEIGHT	LBS
GWT1	GROSS WEIGHT, FUEL WT METHOD	LBS
GWT2	GROSS WEIGHT, FUEL FLOW METHOD	LBS
HDFT	DENSITY ALTITUDE	FEET
HDOT	CLIMB/DESCENT RATE(SLOPE P342)	FT/SEC
IASD	AIRSPEED RATE (SLOPE P002)	KNOT/S
KCAS	CALIBRATED AIRSPEED	KNOTS
KTAS	TRUE AIRSPEED	KNOTS
OATC	CORRECTED TEMPERATURE	DEG C
P1WX	ADJUSTED HORSEPOWER	HP
PCAD	PYLON CONVERSION RATE	DEG/S
PHID	ROLL ANGLE RATE (SLOPE D009)	DEG/S
RHPN	NORMALIZED HP (RSHP/SIGP)	HP
RSHP	ROTOR SHAFT HORSEPOWER	HP
SIGP	DENSITY RATIO	
TDAY	COUNTER START TIME OF DAY	MINUTE
THTD	PITCH ANGLE RATE (SLOPE D010)	DEG/S
TOCG	C.G. FOR RAMP GW	INCHES

The derived parameters for the UH-60 are

Mnemonic	Description		Itemcode
AMU	Advance ratio, derived	( )	(VOMU)
CP	Power coef. (eng. q), derived	( )	(CP00)
CPROTOR	MR power coef. (QMR), derived	( )	(CPMR)
CT	MR thrust coef., derived	( )	(CT00)
FLAP	Corrected blade flapping	(Deg)	(FLAP)
FSCG	A/C longitudinal c.g., derived	(Inches)	(FSCG)
GW	A/C gross weight, derived	(Lb)	(FSGW)
HDB	Boom density altitude, derived	(Ft)	(HDBO)
HPB	Boom press. alt. corrected	(Ft)	(HPBC)
HPS	Ship press. alt. corrected	(Ft)	(HPSC)
LEADLAG	Corrected blade lead-lag	(Deg)	(LLAG)
MTIP	Advancing tip Mach number	( )	(VTIP)
PITCHC	Corrected blade pitch	(Deg)	(PTCH)
REFRPM	Referred main rotor speed	(RPM)	(VRMR)
SHPT	Combined engine shaft Hp	(Hp)	(ESHP)
VCALB	Boom calibrated airspeed	(Kts)	(VCAS)
VT	Corrected compiled TAS	(Kts)	(VTRU)
VTB	Boom true airspeed	(Kts)	(VTAS)

4.2.4.3 Harmonic Amplitude and Phase. The rotorcraft blipper pulse (R018 for the XV-15 and MRTRAZ1 for the UH-60) is used to define the fundamental rotor cycle frequency, then harmonics (6 for XV-15, 15 for UH-60) are derived for selected parameters, using software borrowed from DATAMAP. The harmonic amplitudes and phases are then stored for access in TRENDS via program features HARMONIC, MINMAX, and SEARCH.

4.2.5 Automatic Updating

Supporting and summary files, i.e., files which contain summary lists of all of the counters, itemcodes, and datatypes available for each itemcode, need to be updated as the data are input into the database. Because this was not done in the original TRENDS system, a check of all data in the base was required (as a separate step) each time the database was modified. As the size of the database increased, this step became very time-intensive. Therefore, it soon became necessary to implement routines and procedures for automatically updating the summary and supporting files during the tape- and list-processing operations. These routines are documented in a Procedures Manual which is available from either of the authors.

4.2.6 Record Overflow

The problem of record overflow in a file (because of a fixed maximum allowable record length in the computer operating system for keyed-access files) made it necessary to have a continuation index indicator to connect multiple-record counters. Specifically, only 16,380 bytes may be written to one keyed-access record on the VAX/VMS computer system, which translates to 8150 integer*2 samples/record (part of the record is taken up by overhead; e.g., counter number, scale, bias, start time, samples/sec, number of points). Hence, storage for any mainframe parameters (which are sampled at 251 samples/sec) for any counter longer than 33 sec requires multiple records.

Section V

TRENDS DESIGN/DEVELOPMENT CONSIDERATIONS

5.0 TRENDS DESIGN/DEVELOPMENT

This section addresses some of the design considerations and technical details of the development of TRENDS and gives some of the reasons for decisions which were made.

5.1 TRENDS DEVELOPMENT PHILOSOPHY

The initial philosophy was that the TRENDS system was to enable nonprogrammers and noncomputer people to selectively access and see all data involved with rotorcraft flight testing. The basic goal was to provide not only a user-friendly system, but a complete system.

5.1.1 User's Manual

TRENDS was initially designed NOT to require a user's manual; hence, in-line help was a requirement. It is now recommended that the new user start with narrative information menu items and then work into the plot options. Our advice to the user is, when in doubt, try it. The user can't hurt the system. The user's manual can be useful when looking up parameter names or when running sophisticated applications available in the system. Note that full documentation of this system results in a large user's manual which is believed to act as a deterrent to using TRENDS for the new user. Most of the key features of TRENDS need little or no documentation.

5.1.2 Initial System Software Attributes

Early versions of TRENDS contained most of the basic plotting and searching capabilities of the present version, but the user interface was less formal. Unique special features were imbedded into the syntax of the system initially; however, because of the pressure from the younger engineers, who wanted everything clearly defined, most of this kind of input has been formalized. It appeared that senior engineers were happy to have any means at all for displaying the test data and didn't mind a trial-and-error approach in interfacing with TRENDS. The basic problem was in providing the new user with a user-friendly system while providing the experienced user with special capabilities, yet not complicating the syntax of the input prompts.

5.1.3 Incremental Development Concept

TRENDS' development was user-driven from its initial conception. The priority for work on the DBMS was to

1. Get it working, albeit crudely or incompletely;
2. Respond to users' gravest complaints or requirements;
3. Work on extending/improving system features with user-friendliness always in mind.

As simplistic as this approach appears, it succeeded in making progress while maintaining a useful tool. Fortunately, the initial user group was small and was tasked to solve real problems, not hypothetical ones; hence, only minimal time was spent in trying to justify actions taken during the development of TRENDS. Also, the development of this system involved only four people, including the authors of this report; hence, system direction and continuity have never been sacrificed during the seven years of TRENDS development. The following paragraphs follow the chronological order of system development, as driven by user requirements. For a chronological summary, see Appendix A.

5.1.3.1 Min/Max Plotting. The first plotting capability required for TRENDS was that of plotting one Min/Max item versus another for a specified range of counters. To satisfy the requirement without a detailed design effort, a somewhat general-purpose plot package was provided which allowed the users to input any comments or have any setup that they wanted. Plots were tedious to set up with this package. An early decision was made to sacrifice the general options a user might request and to provide a user-friendly program which would provide labeled, scaled plots on the screen within seconds and require minimal input specifications. Provision was then made to let the user override the labels and scales. Later in the development, the ability for the users to put their own titles and comments on the plot was provided for, but the great majority of the users were and still are satisfied with the titles and scaling that TRENDS provides automatically. It has been found that engineers who are actually doing research are primarily interested in getting results simply and quickly. However, new features for the TRENDS system are given consideration if they don't in any way complicate the input syntax.

5.1.3.2 Narrative. TRENDS represented a major break with respect to existing engineering databases of the time, circa 1982, with the inclusion of as much narrative data into the base as possible along with numerical data. The intent was to allow a user of the system to know what the test was about, and its results, by reading the flight notes and looking at numerical or plotted data. In the design of TRENDS, it was felt that having narrative data in the system would be very useful; however, an appropriate technique for integrating numerical and narrative data was not clear initially. It was the decision to use the test-point number (counter) along with derived counter sets (pseudo-flights) which cemented that integration.

5.1.3.3 Pseudo-Flight Creation. Providing the user with the ability to search through all types of data is normally a standard capability in most databases; however, the ability to search and save test points which satisfy the user's narrative and/or numerical templates semiautomatically on the computer in the user's home directory was the feature which validated the pseudo-flight/derived-flight concept. This concept is crucial to TRENDS and its users, since it allows one to do extensive research on multiple flights easily. Also, pseudo-flight generation by the user personalizes the database to each user's needs; the users in effect build their own databases, but with the unique aspect that they need to save only the test-point numbers in their directories and not the actual data, thereby not increasing storage requirements to any significant extent.

5.1.3.4 Time-History Plotting. This menu option is the most powerful one in the TRENDS system, because most analysis involves plotting of time-histories. TIMEHIST became the home for many options which could be thought of as separate functions except for the fact that they are all involved with the plotting of data. TIMEHIST allows users to input their own functions or to use predefined ones found in a function file. It allows all of the mathematical functions, including table lookups, to be used by simply typing them in the prompt line. It allows reiterative processing of time-history data by executing a process and saving the results of that process with a user-defined file name to be used again. The data in each case are shown on the screen of the user's terminal. Although the integration of so many capabilities into one menu item has at times been challenged, most users generally like the power this single menu item gives them because the prompts can be as simplistic or as complicated as the user requires.

5.1.3.5 Formula Evaluation. It was obvious that an analysis system would have to have the ability to combine data through mathematical operations. Except for spread-sheet programs, this usually requires that a program be modified, recompiled, and relinked each time a new function (or subroutine) is to be included. TRENDS was required to supply users with any functions they wanted and to be able to generate them without having to compile the program. This development was a major effort which required formula-parsing and evaluation techniques in line with searching and plotting procedures to evaluate the functions without compiling. The end result of this software capability allows users to not only generate functions (i.e., formulas) and save them, but also allows users to specify the functions interactively without having previously defined the functions. Most user-generated functions can be used on either statistical or time-history data with no distinction. The function specification and evaluation capability of TRENDS possesses most of the capabilities of FORTRAN formula specification. Conditional statements have not yet been enabled.

5.1.3.6 User-Friendly Autoplot Setup. Autoplot setup, as mentioned previously, is a way to significantly reduce the number of user prompts required to produce a finished-looking plot on the screen. This is accomplished by having special files which are automatically accessed each time a plot is required. These special files supply project titles, test-point descriptions, flight descriptions, and full descriptions of the parameters for the axis labels. Also included in the autoplot setup is an automatic scaling algorithm which provides reasonable scales. Although any part of the automatic information can easily be overridden by the user, most users find the automatic provisions of the MINMAX and TIMEHIST plot features to be assets which reduce the time required to get meaningful plots. Other features in this autoplot setup allow the user to recall previous plot setups which were saved under supplied file names, modify those setups if desired, or if it is found that a mistake has been made in a difficult setup, to easily recall the entire plot setup while still within the plot-dialogue logic (even if the setup hasn't been explicitly saved).

5.1.4 TRENDX

It was found necessary to generate two versions of the TRENDS database operating system in order not to interfere with the user community as modification work was being done on the working software. TRENDS became the released version while TRENDX was the version of TRENDS containing the latest modifications (and made available for acceptance testing by some experienced users).

5.2 DATA PHILOSOPHY

In TRENDS, data reliability imposes a requirement on the user as well as on the database manager. Our aim is to do our best to check the data before they are installed in the base, but speed is of the essence in this decision; hence, if users finds data they think are bad, they should notify NASA. Not all databases are handled in the same manner and there is disagreement among the NASA engineers as to the extent to which flight data must be certified before making them available to the public. In real data systems, errors occur due to bad instrumentation, improper calibrations, bad sensors, broken wires, etc; hence, these types of problems must be considered when using any database. It is extremely difficult and expensive to check the validity of all of the data in a database consisting of billions of bytes, yet there is concern that to make uncertified data available might result in some users writing reports which are invalid and for which NASA could be blamed.

Certain checks are made for obvious errors on all data as they come in; however, subtle errors may not be found in the data unless the user recognizes them. This is where engineering experience is invaluable on the part of the user. One must pay a price to allow data to become immediately available, and that price is the guaranteed reliability of the data. Unlike the XV-15 database, the UH-60 database has been reviewed by knowledgeable engineers specifically to remove bad data. The

UH-60 database is relatively small (500 megabytes) and it took over a year to certify that it was valid. Yet if you ask those who certified the data whether all the data in the UH-60 database are good, their answer is that they are not certain. How long will it take to validate 30 gigabytes of pressure data when they are added to the existing UH-60 database, and can we wait that long? It is unclear how certification can be run on all UH-60 data without spending years in the process. The aim of TRENDS is to interface to users who understand the difficulty of this problem and are willing to help with it. It is felt that users require certified data should use only NASA reports, even if they take years to get, rather than to question the usefulness of a system which is trying to get data out rapidly. TRENDS represents an experimental concept of making raw data available to the users for their own research. The future of TRENDS may be in question if the user community requires that all data be perfect data.

5.2.1 On-Line Data

System speed of response has always been an important consideration in determining user acceptance of a system; hence, there has always been a requirement to provide all data on line within TRENDS (as opposed to having to have tapes or disks mounted in order to access data of interest). Although we still have storage problems, technological improvements in storage, along with data compression, have allowed us to achieve a system capability where all pertinent data are on line via disk storage. Total TRENDS database storage is currently about 5 gigabytes. In an attempt to maintain this on-line capability in a limited floor space and with limited funding available for storage media, NASA has chosen to go to a write-once-read-many (WORM) optical disk storage jukebox system to bring on line another 80 gigabytes of storage. Although there will be slight delays in the new system as compared to the old, it is still expected to be able to provide on-line storage of all data.

5.2.2 Database Management Considerations

The details of creating and maintaining a flight-test database are many and varied. Logistics of acquiring the data from the source must be considered, as must such details as the operational status (and operating system) of the computers which host the databases. Most of these details are abhorrent to aeronautical engineers and not of much interest to programmers. Software and procedures must be developed to automate the process as much as possible. Resource contention and database cleanup are examples of the details of maintaining a database.

5.2.2.1 System Readiness. The activity of filling the database can block the use of the system from the accessing community if some care is not taken to avoid this problem. Special measures are taken to avoid interference with active users during database management operations by copying database files to temporary files, modifying the temporary files, and then recopying them back into the main database. Changes in file formats must also be made in such a way that the users are not taken off line during modification and checkout.

5.2.2.2 Data Deletion. User-friendly data-deletion features which allow new data to overwrite old data are important to the database manager. Inevitably, some existing data in the database will be found to be incorrect. The keyed-access file structure has provided a fairly simple means for manipulation (i.e., filling, deleting) of the data in the base on a nonchronological (i.e., nonsequential) basis. That is, records may be added, modified, or deleted at any time, without concern for which records precede or follow them in the files, thus permitting the processing of current flights of interest first, then using a back-and-fill procedure to complete the database.

5.2.3 Multiple Databases

The TRENDS system was initially designed to support two separate though similar databases for two tilt-rotor flight-test programs. Both of the databases used the same set of itemcodes and the same conventions for test-point specification, although data-source formats were different. The database set was expanded for a short period of time to incorporate some data from the RSRA helicopter program. Support of RSRA by TRENDS was the first attempt to create and access a completely different database. A copy of the existing XV-15 TRENDS code was made and modified to support the RSRA database. This method was quick and led to results in a very short time, but it proved to be a mistake. It would have been better to have taken the time to develop the generic software which would treat the peculiar characteristrics of all of the databases with the same generalized code. The fourth database was for the Phase I UH-60 flight-test program. This ushered in the first attempt to solidify TRENDS into a generic code. A wind tunnel database was also added to the system; however, its impact on the original code was minimal. Note that the integration of new databases into the TRENDS domain has helped to improve TRENDS by making more demands on the system, from which new capabilities become available to all users of the system for all databases supported by TRENDS.

5.2.4 Database Structures

TRENDS is firmly based on the keyed-access file structure and operating method. Most of the numerical data reside in files named by the parameter's itemcode or mnemonic and keyed by counter number. These files reside in directories named for the specific data type they contain. For example, the Min/Max statistics for itemcode D186 are found in a file named D186.MMC which is found (for A/C 703) in a directory named [DB703.MMC]. The record in which the Min/Max statistics for counter 14500 are found is keyed by the integer 14500. The VMS operating system performs most of the overhead in locating the record. This ties TRENDS to the operating system, but it takes advantage of system efficiencies and offers the potential for improvement with system upgrades.

Much of the supporting information, such as which counters are in a given flight, could be found by reading all counter-description files for the flight in which the counter is found, but this would take a long time and waste both the user's and the computer's time. TRENDS creates and uses a number of supporting files to store information which groups and categorizes other data in TRENDS databases. These will be discussed in this section, along with the supporting files which augment the numerical data with descriptive information.

5.2.4.1 Keyed Access. The keyed-access file capability provided by DEC under the VAX/VMS operating system is perfect for a database management system such as TRENDS. If this capability were not available, it would have to be synthesized. The keyed-access capability lets one retrieve a record from a random-access (i.e., disk) storage device by specifying a NAME or NUMERICAL LABEL for the record. For example, the descriptive information for item M143 in keyed-access file ITEMINFO.DAT is retrieved by asking for and reading the record named M143. It is not necessary to read each of the records in the file and to test if the record which was read was for M143 (a process which would be the only recourse if ITEMINFO.DAT were of SEQUENTIAL organization). In the past, when databases were stored on magnetic tapes, leading files and records had to be read (or, with a priori knowledge of the sought data, skipped over) to get to the information of interest. If ITEMINFO.DAT were of indexed-sequential (direct-access) organization, one would have to keep a catalog which associated M143's record with a sequence number, search that catalog to find the appropriate number, and then ask for the record by sequence number. This process would get very complex when new records were added or old ones deleted. With keyed-access organization, the VAX/VMS operating system does the bookkeeping for you, simplifying the programming of deletions, additions, or replacement of records.

It might be argued that direct access should be considered over keyed access for data records for which the key is a positive integer such as counter number. Not only is this a restrictive condition, but direct-access files are observed to have two fundamental problems: (1) they require fixed-length records (and are, therefore, wasteful for variable-length such as time-histories) and (2) empty records take up just as much space as full records, so databases with sparse counter sequences are very wasteful of space. Keyed-access files can handle variable-length records and they do not leave space-holders for empty counters.

Keyed access is operationally convenient for filling and managing a database because the order of filling is of no consequence. That is, later (i.e., higher-numbered) counters may be processed prior to earlier (i.e., lower-numbered) counters.

A very useful feature of keyed-access files in VAX/VMS is that they may also be read sequentially, if desired. In this case, the results are ordered in ascending order of the keys (i.e., numerically or alphabetically ascending), providing a simple means for producing a sorted list. A variation of this feature lets you first read a particular named record and then read sequentially (i.e., ordered) from that record onward. This method of reading in data has been used in parts of TRENDS to optimize the access in the presence of "look-ahead" system disk-reading procedures.

5.2.4.2 TRENDS File Structures. One must consider many factors when choosing a file structure for a database. Some of these factors are:

1. How will the information from the database be used?
2. How much data must be archived?
3. What are the procedural limitations and constraints?

The basic keys for databased numerical (e.g., Min/Max or time-history) records are (1) item name (itemcode, mnemonic, etc.) and (2) counter (test-point identifier). That is, given the data item and test-point index, the data record should be identified for retrieval. Of course, we assume that the database is given and that item and counter are unique within that database. Given the requirements for basic keys, there are still many ways to structure files in a database. Choices for file structures and formats were made early in the design of TRENDS and most of those choices have proven to be good. A detailed discussion of the options may be found in Appendix D.

5.2.4.3 Supporting Files. The bulk of TRENDS' stored information is numerical test data from sensors, but a TRENDS database is completed by another kind of information: that found in supporting files. Supporting files come in two categories: (1) descriptive files and (2) summary files.

Descriptive files augment the numerical test data by providing narrative support such as item descriptions or project, flight and test-point descriptions. Summary files enable efficient searching and use of the counter-keyed numerical and descriptive files in the base by providing precompiled, categorized lists of counters for which data exist in the base. Figure 5.1 illustrates the function of these supporting files in exercising some key menu items.

1. Descriptive Files. These files add information to enhance the use of the numerical data in TRENDS. They include

   a. CTRDESC.A/C, the counter-description catalog file, keyed by counter and containing other counter-reference information such as start/stop times, rotor revs, flight number, and data-group availability

   b. ITEMINFO.A/C, the item-information file for each database, keyed by data-item mnemonic and itemcode, containing item descriptions, data units, mnemonic/itemcode correspondence, and data-group codes

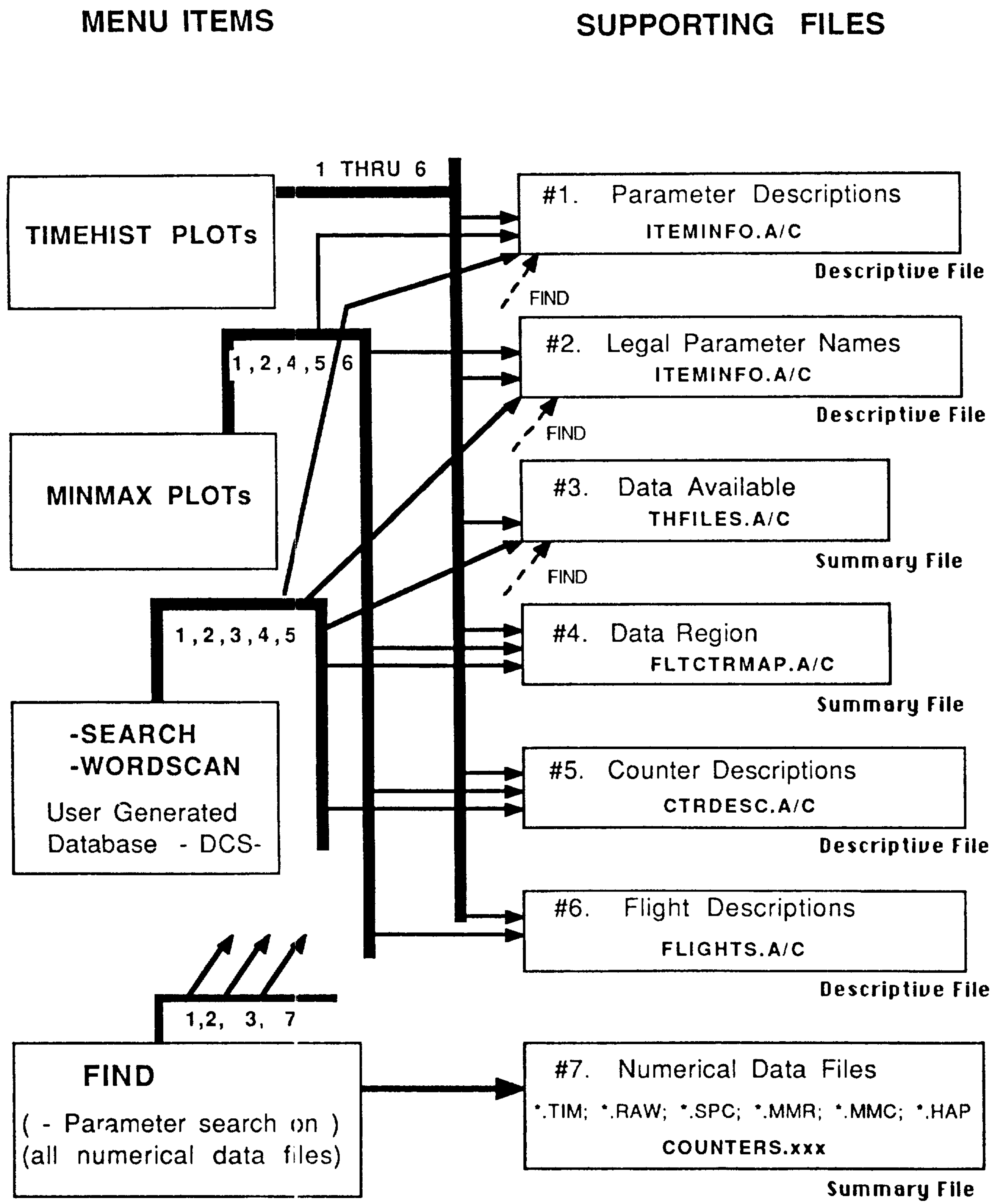


Figure 5.1 Supporting Files & Interfaces

c. FLIGHTS.A/C, the flight-description file, keyed by flight number, containing categorized flight-descriptive narrative information

2. Summary Files. These files contain information which allows efficient access to the database and which enables display of current database status. They include

   a. FLTCTRMAP.A/C, the file which contains the catalog of counters for each flight, keyed by flight number and used to pull up a set of counters when the user asks for a flight

   b. THFILES.A/C, the file which contains the summary of time-history data available in the database for a given parameter keyed by itemcode or mnemonic and containing a list of the counters for which data are stored for each time-history data type or group

   c. COUNTERS.XXX, the files for each data type (XXX), keyed by counter number and containing a list of the names of the parameters for which data are stored for the specified counter and data type

5.3 USER-FRIENDLINESS

User-friendliness was a strong guideline in TRENDS program development. Any new program feature was always designed for robustness and ease of use. Prompts and required responses were studied carefully to make sure they were logical and simple. Because the intended TRENDS users are engineers, rather than programmers who can modify software to get the desired result (given the time to do it), the user interface had to be complete. For example, prompting capabilities had to be included for creation of some user-files (e.g., files for PERFPLOT and CPRINT setups), while programmers could be given the format and asked to write their own.

Among the user-friendly features of TRENDS are (1) the capability for entry of data-manipulative formulas (functions) in plot and search dialogues, (2) a very simple plot-specification syntax, together with the ability to recall and edit stored plot setups, and (3) integration of narrative data to support the stored numerical data.

5.4 USER FEEDBACK

It has been observed that no two users have the same requirements or preferences with respect to database or system operations. We have tried to satisfy a broad range of users by making studied judgements of what is essential, user-friendly and logical. By presenting a system which performs useful services with little effort on the part of the user, TRENDS has attracted a large community of users. If we had catered to a few particular users, TRENDS would probably not have the following it has. User comments and suggestions are solicited and evaluated, but not necessarily followed (or at least not with the priority the user might give the suggestion). Every effort is made to provide special software for special requests for capabilities or data which are not available in TRENDS or its databases.

5.5 ANALYSIS TOOLSET

The accessing part of TRENDS currently consists more than 20,000 lines of executable code and over 200 modules (exclusive of DISSPLA, DATAMAP, GTRSIM, command files, help-files, pointer data, etc.), so it has become necessary to develop and use computerized tools to keep track of software details and dependencies. The use of FORTRAN-source-scanning tools and keyed-access sorting techniques to analyze and summarize the use of subroutines, commons, files and variables is very advantageous when trying to maintain the software of a program the size of TRENDS. Appendix E contains a hierarchy chart or calling tree of the TRENDS accessing software, produced by the toolset.

5.6 INSTALLATION OF TRENDS

TRENDS can be installed on any VAX/VMS system which has the DISSPLA graphics package available. It has been installed at AEFA (with the Megatek graphics package) and at MDHC (on a MicroVAX), as well as on several different VAX nodes at Ames. The source code is written entirely in DEC-extended FORTRAN. The disk storage requirements for TRENDS (i.e., for the accessing program -- not including the database management software) are:

Source	2000 blocks (1 block = 512 bytes)
Object	1400 blocks
Executable	1600 blocks
Help files	1000 blocks.

TRENDS requires a virtual page count of at least 50,000 to run.

TRENDS also provides gateways to the DATAMAP and GTRSIM programs. DATAMAP's executable module occupies about 800 blocks, but the program requires about 6000 blocks of user disk space to execute when scratch-file storage is included. GTRSIM's executable module occupies less than 600 blocks, but the TRENDS interface routines require another 3500 blocks (which requirement could be reduced to about 2000 blocks with more efficient coding).

Any questions regarding access to TRENDS or installation at your facility can be addressed to the authors (Bondi at (415) 604-6341 or Bjorkman at (415) 964-1844).

Section VI

CONCLUSIONS

6.0 CONCLUDING REMARKS

TRENDS has been used within the FA division at Ames for almost seven years as the primary means of archiving and accessing flight test data for rotorcraft. It has recently begun to be more widely applied and used by other Ames organizations and by industry as well. Its capabilities and potential for extension to a wide variety of test data make TRENDS a valuable tool.

This section concludes with a summary of some of the lessons which have been learned during TRENDS' development and an indication of the direction TRENDS might take in the future.

6.1 LESSONS LEARNED

In the development of a large database operating system, one hopes to learn important lessons which might help someone else in development of other database systems. The following items are some we believe to be worthy of mention.

6.1.1 User Impact on the Development of TRENDS

It has been observed that no two users have quite the same requirements or preferences with respect to database or system operations. We have tried to satisfy a broad range of users by making studied judgments of what is essential, user-friendly and logical.

You can never satisfy all of your users; therefore, let only the mature users drive your software system development. Software development to solve hypothetical problems only buys unnecessary complexity and produces essentially no useful output for most of the user community.

It has been found in working with aeronautical engineers that ALMOST meeting the requirement is NOT acceptable to some users. Such users turn off the entire system, not just the part where there is a problem. All of the extras on the system will not bring them back to it unless their essential requirements in ALL areas are met. The lesson is that you can't please all of the people all of the time, nor some of them any time.

6.1.2 Database Evaluation Criteria

TRENDS has been compared to a file cabinet full of reports and plots. It was judged to be valuable only when it could replace most functions of the cabinet. TRENDS was accepted by one "hard-sell" engineer because:

1. TRENDS not only has all of the plots and narrative information available, but it allows the user to search through both narrative and/or numerical data to group it into data sets (pseudo or derived flights). This is not simple with a cabinet system.

2. Accessing miscellaneous plots from different flights with TRENDS is faster with the computer than with a manual system, and the presentation of the parameters on the plot can easily be altered (unlike the file cabinet plots).

3. Mathematical functions can be easily and rapidly applied to parameters through TRENDS, but hardcopy plots never change.

4. When any other (including remote) aeronautical engineers need to see flight data, TRENDS is available to provide it, unlike the file cabinet which requires that the cognizant flight-test engineer always be available.

Note: The ability to store the contents of a file cabinet in a computer system and provide access to that information as readily as one might use a file cabinet is a significant challenge to any systems designer and could be considered as a good benchmark of the performance of any database system.

6.1.3 Access to Calibrations

TRENDS provides access to parameter calibration information (for the XV-15 database) as a main-menu item, CALIBS. Even though the test data are presented to the user in engineering units, access to the calibrations through TRENDS has proven to be a useful capability for many users, especially when trying to determine data validity and causes of some anomalies.

6.1.4 Generic Code

Generic code development is important to the task of maintaining system software. This was not appreciated until the complexity of the TRENDS code increased to the point where it was impractical to maintain three systems which were similar but not identical.

When generic software was initially implemented into TRENDS for the integration of multiple databases on different rotorcraft, it was done very late (about 4 yr into the system design). A significant part of the problem for the generic code was to accommodate the handling of new and longer parameter names, specifically the eight-character parameter names used on the UH-60 versus the four-character ones initially used on the XV-15. Each reference to parameter name information had to be revamped. It is felt that many bugs in the system which had already been corrected resurfaced because of this necessary code change. A ripple effect of system crashes seemed to enter into the TRENDS software as a result of this late generic modification. Current thinking is to try to keep all software generic from the cutset; that is, one fix fixes the same problem for all databases. Special code development should be negated. In the long run this special code will generate more problems than it is worth. If at all possible, do not let the code split.

The following are some areas where generic code development should be considered:

6.1.4.1 Database Vectors. Do not have the location of the database embedded in the code, but rather get it as an input to the program. Initially the path name for the data was embedded in the code, but this created two problems; namely, the program had to be changed if the data were relocated, and when software was put on another computer and had to be modified and recompiled.

Appendix F presents the method used in TRENDS for vectoring to a database.

6.1.4.2 Parameter Names. The logic in a system should not assume a fixed length for parameter names, but should treat variable-length names in a field long enough to take care of all likely mnemonic lengths. To date, a maximum length of eight characters has proven to be sufficient for any TRENDS database.

6.1.4.3 Symbol Table. Symbols for the data parameters should be input from an external file rather than be compiled into the software. This permits flexibility in choosing names, modification of the parameter descriptions, and moving from one database to another without compiling. TRENDS uses a file named ITEMINFO which contains one record for each item. Records are keyed by the item's mnemonic (or itemcode) and contain descriptions, units, instrumentation group and other data.

6.1.4.4 Help Files. The help which is displayed in response to a query by the user should come from input data files created via the system editor, as opposed to being compiled into data statements in the program. This gives flexibility and reduces the effort required when the information changes.

6.1.4.5 File Structure. Make common file structures for similar files in each database, e.g., Min/Max statistics files for XV-15 and UH-60 are structurally identical even though the UH-60 has more statistical measures in each data record. Time-history file structures are are identical for all TRENDS databases.

6.1.4.6 Data Types. TRENDS' time-histories were categorized into data types (e.g., *.raw; *.spc; *.aer) for accounting purposes. This proved fortunate because it simplified the dispersion of data files onto different disks when we ran out of storage on the original disks. The initial time-history file format in TRENDS was one file per parameter for all flights; the ultimate limitation of such a system would be one file filling an entire disk. Before that occurred, however, it would be necessary to migrate parameter files onto other disks transparently.

A different file format, the DATAMAP and TRENDS (DAT-TH) format, will be used in the future to help to alleviate the problem.

6.1.5 Plotting Generality

It is better to provide a specialized plotting format which is easy to specify than to provide great generality and graphics capability. The types of plots enabled by TRENDS cover most of the needs of Ames engineers. Users can specify non-default labels, scales and titles with a little extra effort and can store/recall plot setups to avoid re-entering specifications. TRENDS works with a variety of different terminals.

6.1.6 System Response Speed

User acceptance does depend on the system's response and system response is therefore a key consideration when writing code. Two of the techniques used in TRENDS to achieve better response times are

1. Summary support files, updated automatically during data-fill, to eliminate the need to query the whole database to see which counters and itemcodes and data types exist

2. Keyed access to locate the first data in a requested sequence and then sequential reads with tests (uses efficiency of system look-ahead algorithms)

Note: Summary support files are in fact data-redundant; however, the increase in file storage is insignificant relative to the improved response of the system. These summary files must be automatically updated every time the main database is updated.

6.1.7 Database Management (DBM) Menu

Getting data into any database system is not a trivial task; the TRENDS system manager therefore decided to approach the problem formally by creating a data input menu system by which all data are input into the database. To a minor extent, each DBM menu is tailored to each rotorcraft's database. The current TRENDS DBM menu is not complete; however, it is being used by database management personnel at Ames and it is felt by the authors to be an asset. Although there was disagreement about the need for such a menu and its development was given a low priority, this type of software is important, and can be critical if the program is transferred from one contractor or management team to another. This software provides the TRENDS manager with better control for supporting his system if he has to introduce a new team to a TRENDS data-input task. Finally, this type of software helps experienced users avoid mistakes.

6.1.8 Flight-Test Support

TRENDS can be used as a flight-test-support tool, as it was in the Phase I JH-60 tests at AEFA. In this role, the narrative stored in TRENDS can be fresh and accurate and used to document the tests. Data anomalies can be spotted and instrumentation/reduction errors can be corrected before they permeate the entire test series. Data requirements (such as the need for measured rotor azimuth) can be noted and possibly satisfied during the testing.

6.1.9 Data Quality Considerations

Concern for data quality imposes a requirement on the user as well as on the database manager. Automatic and spot checks will not identify all of the of the errors in the stored data. Users must report problems to the database manager for resolution. It is felt that users who must have completely certified data should use only NASA reports and that TRENDS users, while they can be reasonably confident in the data stored in the base, should be realistic and watchful.

6.1.10 Data Storage Considerations

Disk storage is probably the most important of all considerations for flight-test databases. If there are trade-offs between execution time, convenience etc. and storage in the design of a system, the edge must be given to minimizing storage. One significant decision in this regard led to the storing of all time-history data as two-byte integers rather than as floating-point numbers in engineering units; another was to store only start-time and time increments between samples rather than to store time with each measured data sample.

6.1.11 Time and Time Offset

Data samples recorded in a flight-test system are usually recorded at regular intervals of time. This makes it acceptable to archive only the starting time and the time increment (or equivalently, the sampling rate) with each time-history record. Not all time-series begin at the same instant, however, and not all time-series for a specific aircraft, instrumentation system, and test point have the same sampling rate. Therefore, to enable detailed analyses, great care must be taken to gather and store the time parameters in the right way. In particular, mainframe and subframe offsets must be obtained and stored so that relative phasing of flight parameters is preserved. The starting time for XV-15 data was stored in single-precision, floating-point seconds for the tens of thousands of test points in the database. The significance of the mantissa in the VAX floating-point word is only about 6+ decimal digits, so the resolution of the starting time is only about 0.1 sec and therefore not very useful in rotor phasing calculations. It would have been much better, in retrospect, to have stored the starting time in four-byte integer milliseconds or to have stored seconds modulo 100 for example. To correct this problem now would require reprocessing thousands of magnetic tapes.

6.2 LASER JUKEBOX CONSIDERATIONS

In 1990 the TRENDS system will incorporate a 90-gigabyte laser-optical jukebox system because of anticipated requirements for flight-test data storage. The time required to access time-history data will increase from what it is currently (on fixed, magnetic diskc), especially when the jukebox must change platters (i.e., optical disks) to get to the data. It is uncertain at this point whether or not some sort of caching scheme will have to be implemented in order to adequately support multiple users and/or to improve time-history viewing performance. A 5 1/4" (disk-diameter) system was selected because it appeared to be much more cost-effective (both system and media-wise), than a 12" system. The 5 1/4" system requires much less room (less than one standard five-drawer file cabinet), uses standard 15-amp, 110-volt power, and appears to be less complicated (hence, more reliable, we hope) than 12" systems. The use of WORM rather then erasable optical disks is based on the archival character of the data to be stored and the WORM's high reliability, which does not require tape backup.

The system being acquired is a 90-gigabyte Mitsubishi Laser Optical Jukebox library system (Model MW-5G1-B4) with four (4) optical drives (model MW5D1), and two (2) KOM SCSI/UNIBUS controllers interfacing with KOM Optiserver (Version 2.x) and Optifile II software for the VAX computer.

6.3 FUTURE TRENDS FEATURES

One piece of information which is not included in TRENDS (but which would be helpful) is the location of all transducers (i.e. sensors) on the rotorcraft. To provide an accurate computer-graphic picture is complex and in some instances would require a number of drawings with enlarged sections. This feature is being considered as an additional desirable capability.

The further integration of mathematical model simulations into TRENDS is high on the priority list for future expansion. Some of the simulations being considered for integration are CAMRAD, CL81, GENHEL, TILTWING and NASTRAN.

6.4 FUTURE OF TRENDS

Finally, the biggest concern is how to support TRENDS through the years as NASA organizations change, user support changes, funding shortages occur, computer operating systems change, and computer mainframes change. Maintenance of such a system requires perseverance, vigilance, and concern, but its value as a tool for the engineer is well worth the effort required to to provide it.

Glossary

A/C	Aircraft
A/C 702,703	The two XV-15 aircraft (702 and 703 are the tail numbers)
AEFA	Aviation Engineering Flight Agency, U.S. Army flight-test facility at Edwards AFB, CA
AMA	Analytical Mechanics Associates, Inc., Mountain View, CA
Ames	NASA Ames Research Center, Moffett Field, CA
ARC	Acronym for Ames Research Center
Bell, BHT	Bell Helicopter Textron
BV-360	A helicopter made by Boeing-Vertol of Philadelphia, PA
CCF	Central Computer Facility at Ames
CDC	Control Data Corporation, computer manufacturer
Compressor tape	A digital test-data tape format used at Ames
Counter	A number which identifies a specific test point (sometimes used to mean the test point itself)
Crows Landing	NASA flight-test facility at the Crows Landing Naval Auxiliary Landing Field
Data item	A parameter, variable or channel for which data may exist in a database
Database	A collection of numerical and descriptive information accessible via TRENDS
Data region	The flights, pseudo-flights or counters of interest when exercising a TRENDS search or display feature
DATAMAP	A rotorcraft-data analysis program built for the Army by Bell Helicopter Textron (Data from Aeromechanics Test and Analytics -- Management and Analysis Package)
DAT-TH	New DATAMAP and TRENDS time-history file format
DBMS	DataBase Management System
DCMPRS	A TRENDS database management program for reformatting compressor-format tapes or disks

DCS	Derived Counter Set (See Pseudo-flight)
DEC	Digital Equipment Corporation of Maynard, Mass., maker of the VAX computers
Derived items	Data items not directly measured, but derived from other data items for storage in a TRENDS database
DISSPLA	A graphics package, product of ISSCO, Inc. of San Diego, CA and widely used on VAXs at Ames
DNW	Deutsch/Nederland Windtunnel facility in The Netherlands
DTF	Data Transfer File format acceptable to DATAMAP
EDF	Engineering Data File, a data format produced by the Ground Data Center at Ames
FA	Aircraft Technology Division at Ames (FAF and FAR branches are principal TRENDS users)
FFT	Fast Fourier Transform, a method for determining the spectral (frequency) characteristics of a signal from a time series representation
Flight	An identifiable portion of a test project, such as one day's test results or a run (batch) of windtunnel test points
FOX,FOX4	VAX computers at the Ground Data Center at Ames
Generic code	TRENDS software which applies to all TRENDS databases
Gigabyte	One billion bytes
GTRSIM	The Generic Tilt-Rotor Simulation program, developed by Bell Helicopter Textron and modified by STI
Harmonics	Fourier coefficients which synthesize a signal's frequency characteristics at multiples of a fundamental frequency (i.e., the rotor rev rate)
HARP	A wind-tunnel database for the Hughes Advanced Rotor Program (now an MDHC program)
In-line function	A defined formula or function evaluated during execution of TRENDS
Itemcode	A 4-character code which identifies a data item or parameter (the original XV-15 itemcodes had one letter and three numbers such as M143 or P002) (See also Mnemonic)

Keyed access	A file-access attribute, available on VAX/VMS, by which a record of data is identified by name (as opposed to direct access, where the record is identified by sequence number)
Laser-optical	A new technology for data storage
LOTUS 1-2-3	A software product (spreadsheet) of Lotus Development Corporation of Cambridge, MA
MDHC	McDonnell-Douglas Helicopter Company in Mesa, AZ
Menu item	One of the TRENDS features invocable via the TRENDS menu
Min/Max	Data-item statistics based on the minimum and maximum values observed on each rotor revolution (sometimes used generically to mean scalar or non-time-history measures)
MMC, MMR	Min/Max per counter, Min/Max per rev data types
Mnemonic	A brief name for a data item, such as LONGSTK, limited in TRENDS to 8 characters (See also Itemcode)
NEP	Designation for the Neptune VAX computer at Ames
OFI	The branch for the Ground Data Center at Ames
On-line	Ready for immediate access (an on-line database requires no special tape mounts or special processing to enable access to stored data by the user)
PC	Personal computer (as opposed to a mainframe computer)
Performance	Data items or test results relating to the performance of the test vehicle
Pseudo-flight	A collection of related test points constructed by the user
Rev	One revolution of the rotor of a rotorcraft
RSRA	Rotor Systems Research Aircraft, a modifiable rotorcraft built by Sikorsky Helicopter of Stratford, CN and operated by Ames
RTM tapes	Real-Time-Merge digital data tape format used at Ames
STI	Systems Technology, Inc. of Mountain View, CA
Supporting files	Elements of a TRENDS database which do not contain recorded or derived test data, but aid in the implementation of the system

Summary files	Elements of a TRENDS database which summarize various aspects of the database
TABLES	The name of a Tektronix (and later VAX) Basic program which was used at Ames before TRENDS to plot and list XV-15 Min/Max data
Test point	An identified portion of a test for which data are collected
Test-point index	A number which identifies a specific test point
Time-history	A set of data sampled at a series of sequential time points to show time variation
T-H	Abbreviation for time-history
TRENDS	The TRENDS interactive database operating system (the acronym originated from Tilt Rotor ENgineering Database System)
UH-60	The Blackhawk helicopter, built by Sikorsky Helicopter of Stratford, CN
VAX	Class of virtual-memory mainframe computers, product of Digital Equipment Corp. (DEC)
VMS	Computer operating system used on the VAX computers
VSN	Volume serial number (identification for magnetic tapes)
WORM	Write-Once-Read-Many attribute of laser-optical disk storage
XV-15	Experimental tilt-rotor aircraft built by Bell Helicopter Textron, Inc. of Ft. Worth, TX

REFERENCES

1. GENERIC TILT-ROTOR SIMULATION (GTRSIM)

NASA CR-166535: USER'S AND PROGRAMMER'S GUIDE, SEPTEMBER 1988

NASA CR-166536: A MATHEMATICAL MODEL FOR REAL TIME FLIGHT SIMULATION OF A GENERIC TILT-ROTOR AIRCRAFT, SEPTEMBER, 1988

NASA CR-166537: DEVELOPMENT AND VALIDATION OF A SIMULATION FOR A GENERIC TILT-ROTOR AIRCRAFT, APRIL 1989

2. THE DATA FROM AEROMECHANICS TEST AND ANALYTICS - MANAGEMENT AND ANALYSIS PACKAGE (DATAMAP)

USAAVRADCOM-TR-80-D-30A: USER'S MANUAL, DECEMBER 1980 (RICHARD B. PHILBRICK, BHT)

USAAVRADCOM-TR-80-D-30B: SYSTEMS MANUAL, DECEMBER 1980 (RICHARD B. PHILBRICK, BHT)

NASA TM 100993: DATAMAP UPGRADE VERSION 4.0, MARCH 1989 (MICHAEL E. WATTS AND SHABOB R. DEJPOUR)

APPENDIX A

==

TRENDS Development Tasks Listed in Chronological Order

==

November 1982
  Fatigue analysis at BHT

December 1982
  Design initiated
  Data entry
  Use of TABLES software

January 1983
  Minmax reformatting
  Hard-copy plots on the CDC 7600 computer
  Database summary
  Flight descriptions
  Search on numerical data
  Display of itemcode information
  Quasi-operational ACCESS (TRENDS)
  Study of DATAMAP

February 1983
  Operational ACCESS (TRENDS) software
  User help on line
  Instrumentation groups in ITEMDEFS
  Derived parameters (initial set)
  Reading 702 descriptive data
  TABLES: 2nd-level tables, absolute value,
          other terminals accommodated

March 1983
  Store/recall condition masks in SEARCH
  Show user's DCS and MSK files
  Narrative categories in FLIGHTS
  CHARSORT for listing counter keywords
  Start/stop time added to counter information
  VIEW reproduces listings from stored data
  Time-history storage, harmonic analysis

April 1983
  Manual entry of and access to user's MMC values
  CNTR as MINMAX abscissa or SEARCH variable
  News available to system users
  Extraction of and access to multiplexed temperatures
  Selective data-copying, batch reformatting
  Databasing of one-line flight descriptions for 702

May 1983

- Plot setup files saved/recalled
- LOGSCAN implemented
- CTRL-C interrupts treated internally
- Bending moments displayed vs. blade station
- BHT minmax and time-history tapes processed
- RESCALE option for plots
- Determining column selects points to plot

June 1983

- Time-history plotting
- Minmax-per-rev plotting
- Specific counters excludable from plots
- Plot-data tabulated to a file (PRINT)
- Itemcode descriptions searchable in ITEMDEFS
- Strings of numbers may be input for data region
- Database summaries from database, not canned files

July 1983

- Plotting: 5 plots/page
  - dual, independent plot y-scales
  - single label for multiple same abscissae
  - determining column for single-curve plots
  - choice of curve symbols
- Checks on user-response validity (user-friendliness)
- Reformatting Tektronix data back into VAX for database

August 1983

- Derived parameters include counter duration, computed rpm, gross weight
- Filling database with narrative and data
- Weiberg's gross-weight-by-counter entered into database
- Conversion of hard-copy plots from 7600 to Cray

November 1983

- Initial interface of DATAMAP with XV-15 database
- Summary-file generation (FLTCTRMAP.KEY)

December 1983

- Early work on database-to-DTF for DATAMAP
- Ground-run numbers accepted as well as flights

January 1984

- BHT standard-label to DTF reformatting
- DATAMAP file-creation and processing integrated
- DCMPRS selects items to store and decimates
- Prompting program for flight-log entry

February 1984
- Rotor-blipper extracted from control word, stored
- DCMPRS reads directly from tape, bypassing MTREAD
- Work toward command files for DCMPRS, BELMNMX
- Database status-monitoring efforts
- Backup procedures for database established

March 1984
- Filtering of time-histories in DCMPRS
- Truncated time-histories in DCMPRS

April 1984
- BELBAT for filtering, decimation, selectivity in processing 702 time-history tapes
- Filtering algorithms and convolution implemented and tested
- Separation into separate databases (disks, directories) and separation of software from data

May 1984
- Subdirectories by data type (e.g. [DB703.MMR])
- MMR plotting
- Manual entry of list of counters to usable DCS
- Deletion, replacement, edition, addition of flights in BASKER
- Hangar runs accepted as well as ground and flight
- Experimental (ACCESSX) software used in development

June 1984
- Acceptance of group names (e.g. AER) in DTF creation program for DATAMAP
- Gross condition checks made on MMC database
- Database-management tools improved
- Numeric listing in ITEMDEFS as well as alphabetical
- ADDCOM tool for data-entry by NASA personnel

July 1984

- Comparisons of MMC from DCMPRS with those from FOX
- COPYFLTS minimizes file interference with users
- Maintenance program integrated into DATAMAP
- PRINT with options in TIMEHIST
- FILLMAT3 off-line database-access routine supplied
- Cross-checks for MMR, MMC vs. time-histories made

August 1984

- New plotting dialogue in TIMEHIST
- Creation of SPC, RAW directories
- New flight-counter-key files created for summaries
- Automatic updating of summary files in DCMPRS
- Special keys devised for long time-history records to handle the record-size limitation

September 1984

- Math operations enabled in TIMEHIST
- Either-or and neither-nor enabled in WORDSCAN
- Interpolation to constant sample rate in DTF allows cross-plot and frequency response in DATAMAP
- Time-history group availability shown in WORDSCAN

October 1984

- Min/Max plotting dialogue updated to match TIMEHIST's

December 1984

- Reverse-chronological listing in LOGSCAN
- Show HQ,AER, etc. in WORDSCAN
- Interrupt WORDSCAN to show items stored for each group
- Utility routine, WHATDATA, scans any TRENDS item-file

January 1985

- Utility software for database review
- Database management procedure automation
- Filtering of time-histories before storing in DCMPRS
- New ACCESS (TRENDS) menu
- TABLES and DATAMAP available via ACCESS (TRENDS)
- HELP capability added
- TERMINAL capability added
- FLIGHTS gives one-liners as well as full descriptions
- FINDTHC reads and produces derived counter-sets
- DCS and F+number accepted at data-region prompts
- Multiple counters recognized for TIMEHIST data region
- MMR data plotted in TIMEHIST
- Polynomials calculated and plotted in TIMEHIST
- Time-to-next-inspection included in 703 one-line flight descriptions

February 1985
- New dialogue for MINMAX, including formulas, like TIMEHIST
- Functions (formulas) in SEARCH
- Pre-defined FUNCTIONS feature, evaluation
- Use of derived pseudo-items in MINMAX functions
- Math-library routines (e.g. SIN) in functions
- KEYS, VIEW, WORDSCAN updated to use DCS, :+cntr, etc.

March 1985
- Change aircraft-of-interest at any data-region prompt
- SEARCH efficiency improved with file re-use
- Climb-rate derived and stored for all P342 time-histories
- Harmonics studied, computed, stored, accessed

April 1985
- 702 harmonics tapes processed/analyzed
- Setup files developed for re-processing old PCM tapes requested by BHI
- Derived items in DCMPRS
- Hardcopy print in DATAMAP
- K notation (for 1000s) for plot scales implemented
- Harmonic MMC type used in expressions
- HARMONIC menu-item for plotting harmonics vs. MMC expressions
- OSC and MAX statistics made usable in formulas
- Comparison of MMC measures from various processing sources

May 1985
- Systematic procedure developed for processing requests
- Plotting of amplitude spectra in TRENDS
- REVDATA menu-item for loads display
- FINDTHC uses THFILES.CAT to display summary of available counters for time-history groups
- Printing of amplitudes/phases in HARMONIC
- M-scaling (for 1,000,000s) implemented for plots

June 1985
- Hardcopy plots available as option
- "Groups" specified by name (previously by number)

July 1985
- New database/disk configuration (FHT2 for 702, FHT3 for 703)
- QWIKPLOT completed
- Automation of production of THFILES.CAT, harmonics, slopes
- Accessing of some RSRA data

August 1985
- RSRA TRENDS

September 1985

- Gross weights reviewed/improved/used in deriving parameters
- SEARCH and MINMAX recognize .SMO, .CMN, .CMX, .FSC measures
- Label, decimal places, units specifiable in SEARCH
- DATAMAP cross-hair feature activated
- Stanford seminar preparation and presentation

October 1985

- MULTIPLT added to menu
- GTRSIM integrated with TRENDS
- ITEMDEFS expanded to include pseudo-items

November and December 1985, January 1986

- INTERVAL in TIMEHIST
- New-version *.LIS files with automatic purging
- HELP/ALL for printable help
- REPORTX developed for scanning/plotting ACSYNT (aircraft synthesis program) data, including contour and families of plots

February 1986

- FILES added, with keyed-access, reserved-name files replacing multiple user-files
- Temperature-scan 703 data available for MINMAX & SEARCH
- Elimination of the DTF step for DATAMAP from TRENDS
- RSRA time-history data available/plottable
- Generalization of software for multiple databases
- Interface between TRENDS databases and Tischler's software

March 1986

- CONVERT (VMS) used to compress databases
- Integration, differentiation, convolution filtering enabled
- PRINT in TIMEHIST generalized to produce ASCII files
- SAVE (now STORE) for saving/recalling derived time-histories
- Table lookup entry and use enabled in formulas
- Modularizing of TIMEHIST for better configuration management
- New RSRA minmax format treated
- User's Reference Manual delivered

April 1986

- Scanned temperature data stored like other MMC data, eliminating the need for a special TEMP file
- FLIGHTS.IND (keyed-access) replaces six direct-access files and extra processing steps
- POLY coefficients displayed in plot legends
- Command file for routing hardcopy plots to plotting devices
- FINDTHC (*) lists all stored items of specified type
- Modularization of subroutine LABELS

May 1986
- TIME accepted in formulas in TIMEHIST
- Blanks and E-notation enabled in formulas
- ITEMDEFS allows backing up to earlier prompts
- RESCALE at data-region prompt in MINMAX and TIMEHIST
- Preparation of off-line module for directly accessing TRENDS time-history data by an external program
- Radar tracking position and velocity data processed from RTM tapes for 703 and accessible in TIMEHIST

June 1986
- CALIBS accesses based calibration data for 703
- BASKER modified for new FLIGHTS.IND format and to accept sequential-formatted ASCII as input
- Support of laser-optical disk demonstration
- Search of full text in FLIGHTS, storage and use of DCS
- Simulation of frequency sweeps for TIMEHIST and SPECTRA
- NOTCH filter implemented in TIMEHIST

July 1986
- FILES accesses other users' files with DIR, TYPE, COPY
- Simulation (GTRSIM) plotting brought into correspondence with other TRENDS (formulas, POLY, scaling, PRINT)
- Key numbers assigned to GTRSIM run results for databasing

August 1986
- Modularization of DISPLT, preparation for TEMPLATE
- Processing of sample AEFA compressor data

September 1986
- ITEMDEFS groups expanded to include temperatures, radar, sidestick controller
- Adaptation of DISPLT to TEMPLATE at AEFA (beginning)
- Updating of RSRA TRENDS

October 1986
- NORMALIZE option, work with TEMPLATE and AEFA

November 1986
- RAW data moved to a different disk from TIM for 703 (first segmentation of a TRENDS database)
- Accommodation of AEFA formats, procedures, computers

December 1986
- Procedures and User's Manuals delivered

January 1987

Utility software for correcting bad data in support files
Radar data from RTM tapes compared with same from EDF
EDIT capability developed for MINMAX and TIMEHIST

February 1987

SAVE/RECALL of plot setups with EDIT
On-site flight-test support for UH-60 at AEFA
Rev-data processing installed for UH-60
Software for comparing TRENDS stats with AEFA's
BIT10 (Boolean) added to library functions
Database management (filling) tasks performed by NASA
(first database management by other than AMA)
UH-60 data accessible at ARC with TRENDS (as well as AEFA)

March 1987

GTRSIM updated to latest standalone
GTRSIM/TRENDS user documentation written as appendix to
TRENDS User's Manual
Software analysis toolset applied to GTRSIM
DATAMAP accepts UH-60 data

April 1987

GTRSIM interface modified
Trip to Boeing-Vertol to demonstrate TRENDS
Database management procedures and programs developed for
handling dual AEFA/ARC software and data for UH-60

May 1987

Reading and storing of weather variables from Crow's RTM
tapes of radar and XV-15 data
Cross-hair point-saving and storing to a file
TITLE override of plot headers enabled
GTRSIM: maneuver simulation interfaced/checked
better entry format for control positions
printing and plotting of aerodynamic tables
Time-jump accommodation work on UH-60 data
TSHIFT capability for shifting time-histories relatively

June 1987

Move TRENDS database to new NEP computer
New calculations for gross weight
GTRSIM's aero tables labeled, plotted, hardcopies delivered
DATAMAP modified to use UH-60 blipper, enabling harmonic
analyses and display vs. rotor azimuth

July 1987
- Accommodate new derived-parameter routing at AEFA
- Compute and store loads data for UH-60
- PERFPLOT developed
- Database-management menu developed for UH-60
- HARMONICS activated for UH-60
- LOGSCAN, DERIVED enabled for UH-60 TRENDS

August 1987
- Efforts towards generic TRENDS
- Crow's landing weather data added to 703 database
- FILLER-NEP (UH-60) modified to store harmonic data
- New time-history format designed and tested for counter-based files rather than item-based files
- Cross-hairs pickling in MINMAX as well as in TIMEHIST
- Expression ditto (# or #X) to replicate complicated entries
- Scale ditto (") to force same scale for two y-curves
- INFOFILE specification prompt for TRENDS' DATAMAP
- GTRSIM operational on new NEP
- Contour plots enabled off-line for GTRSIM

September 1987
- TRENDS demo at BHT
- Laser juke-box tests carried out at Perceptics
- USAGE developed for logging/displaying TRENDS usage
- Menu for GTRSIM rather than list of two-character keys
- Pickling of control histories to be accepted by GTRSIM

October 1987
- Consolidated TRENDS software on NEP1 disk
- Gross-weight-by-counter investigation
- Spike corrections
- Comparison utilities developed for UH-60
- TRENDS/TRENDX experimental version and procedures used
- Saving/recall of an editable ASCII mask in SEARCH for XV15
- MULTIPLT in UH60 TRENDS menu
- GTRSIM: COMPARE to superimpose time histories
  - PERFPLOT
  - Inputs from flight data
  - Correspondence list of flight/sim mnemonics
  - OUTSUB modularizes output statements in GTRSIM

November 1987
- Monitoring and summary of spiky or band-edging data
- Increase SPC time-history samples stored to 1025
- FILDERIVE modified for better gross reasonableness checks on numbers used in derivations
- FINDTHC: HAP added, question mark enabled for all items
- SELECT option added to WORDSCAN
- BENDING reactived off line for ATB and pre-ATB data
- TRMP option in GTRSIM (series of trims from flight DCS)
- Minmax comparison (printed) for GTRSIM flight/sim
- Databasing of minmax and time-history GTRSIM results

December 1987
- Spike-removal efforts, de-spiked raw-data storage/access
- De-spike routine developed
- AND and NOT.AND enabled in WORDSCAN
- TRENDS' GTRSIM matches results of standalone version
- Graphical comparison of trim solutions for GTRSIM flt/sim
- Automatic, overridable collection of flight data for input to GTRSIM

January 1988
- HARP wind-tunnel database added
- Comparison of short/long/old/new UH-60 minmax stats enabled
- Software analysis tools extended/updated and applied to TRENDS, GTRSIM and DATAMAP
- Database management menu for XV-15 begun
- Database location for TRENDS comes from a file at run time rather than being hard-coded

February 1988
- Work toward automatic production of database status report
- TIMEHIST help menus and help text-files developed
- Utility developed to compute minmax stats from time-histories
- SUBINFO summary added to software analysis toolset

March 1988
- BV-360 wind-tunnel database added
- SEQN (sequential abscissa) enabled for MINMAX plots
- List-file logging and printing option at exit from TRENDS
- Modularization of production of list files in TRENDS
- New time-history format and special TRENDS to read it
- Detailed HELP menus for MINMAX, like TIMEHIST's

April 1988
- Revision of 703 instrumentation groups
- Specification of source and destination for UH-60 FILLER to alleviate storage problems
- Derived rotor parameters at mainframe rate for UH-60
- NUKER tool for UH-60 database managers to excise counters

May 1988
- MOVER, FLAGGER tools developed for UH-60 database management
- PROJECT menu-item and information in database
- Testing of a laser optical disk with TRENDS

June 1988
- EDF format converted to TRENDS format
- User's Reference Manual (slanted to UH-60) delivered
- Presentation and participation at UH-60 Workshop

July 1988
- HP2623 terminal-use enabled
- CPRINT (custom print) added
- XV-15 project information accessed by PROJECT
- Main-rotor azimuth available as abscissa in TIMEHIST
- TRENDS access on a PC-AT (via modem) enabled

August 1988
- New time-history format devised, tested and accessing modules written

September 1988
- Time-skew investigations/solutions for 703 data
- Wild-card (*) responses accepted for most data regions
- Generic KEYS with custom KEYITEMS for XV-15
- Condition mask file extended by tail number

October 1988
- XV-15 item groups expanded to include new blade parameters (ATB)
- Accommodation of Cobra data
- BASECOM distributes database parameters, relieves hard-coding
- Demonstration of TRENDS/GTRSIM interface to NASA personnel

November 1988
- Chinook and Apache databases on FSD VAX

December 1988
- Test version of unified TRENDS is operational
- User-generated files (e.g., FUNCTIONS) uniquely identified by database
- Prototype screen-managed menu developed

January 1989
- Formatting of MDHC test tape to TRENDS format
- Training and instruction for wind-tunnel software developer
- Database filling for Cobra data

Release of unified TRENDS
Release of TRENDX with a screen-managed menu
Development of a UH-60 database in counter-file format, access by TRENDS
Development of a utility for page-numbering the TRENDS Report and concurrently updating a table of contents with the page numbers

February and March 1989
Delivery and installation of TRENDS at MDHC and the 40 x 80 wind tunnel
Training/presentation of TRENDS at MDHC and the wind tunnel
Moving of database management software from DBMGR data disk
V22 database installed (project data only)
XV-15 engine model data reformatted and printed
BO-105 database installed on FSD VAX
Unification and improvement of the database management software
Identification (date, time, name) included in user-generated files
Project information enabled for all databases
TRENDS generalized to access either old or new time-history formats

April 1989
Correction of switched (and correspondingly miscalibrated) data items in the UH-60 database
Tagging of user-generated files with database extension
Rotor-blipper signal synthesized for UH-60 flight 25
Unification of ITEMINFO treatment, improvement of wild-card in ITEMDEFS
Improvement of wild-card selection of data items in VIEW
COMPARE improved and enabled for time-history comparisons between databases
Plot titles and database symbols moved to the descriptor file
Apache database on NEP written in new counterfile format on laser disk
Extension of counterfile format to treat floating-point e.u. data
Improvement of the GTRSIM iteface, including s/w configuration control

May 1989
TRENDS installed on KRY VAX, along with the HARP database
Itemcode, as well as mnemonic, recognized in FIND
Scanning and display of contour and family plots for CAMRAD

June 1989
Upgrading of TRENDS on MAR for quick-looking at XV-15 data
FREQN installed to enable "per-rev" as abscissa in spectral plots
Improvement of the software analysis toolset, FATS
Initial steps for accessing Phase II Blackhawk data on FOX4

July 1989
Efforts toward processing new (~bad) XV-15 ground-test data on MAR
Development and test of database filling and management software on FOX4
TRENDS installed on FOX4 (required linking to accommodate new DISSPLA)
Paper prepared and presented at Measurement & Instrumentation Workshop
Work toward unification of the counter-description file throughout TRENDS

APPENDIX B

SYNTAX FOR FORMULAS IN TRENDS

TRENDS provides the user with a capability for combining the stored numerical data according to his own formulas for the purpose of searching or plotting. These formulas may be entered at prompt-time or stored as named "functions" and recalled by name (see menu-item FUNCTION). They may be applied to either Min/Max (scalar) data or to time-histories. The general form of the mathematical expressions understood by TRENDS is:

operand {operation operand} {operation ... operand}

where the operations are any of +, -, *, /, ^ (^ is exponentiation). The operands are either:

* itemcodes or derived itemcodes (with extensions, for **minmax**),
* literal numbers (E-notation accepted),
* names of previously defined formulas (functions),
* library functions with math-expression arguments,
* previously defined univariate table name with math-expression argument
* "TIME" (for time-history plotting only)

Parentheses may be used in the mathematical expressions to clarify the computational order. The default order of computation is left-to-right as encountered (reverse Polish notation). For example,

M143+M107/2 is equivalent to (M143+M107)/2

rather than to M143+(M107/2) as it would be in FORTRAN. All literal numbers are used as REAL*4 floating-point, whether or not the decimal point is specified.

It is important to note that the first field MUST be an OPERAND and NOT an operation. Therefore,

-M143	is invalid, while
-1*M143 or 0-M143	are valid expressions.

Library functions

The available library functions are REAL*4 functions of a single REAL*4 argument, X, which may itself be a mathematical expression.

```
SIN(X)       sine of angle X in degrees
COS(X)       cosine of angle X in degrees
TAN(X)       tangent of angle X in degrees
ASIN(X)      arcsine of X, returned in degrees (-90,90)
ACOS(X)      arccosine of X, returned in degrees (0,180)
ATAN(X)      arctangent of X, returned in degrees (-90,90)
SQRT(X)      square root of (absolute value of) X
EXP(X)       exponential of X
LOG(X)       logarithm (base 10) of (absolute value of) X
LOGE(X)      natural logarithm of (absolute value of) X
ABS(X)       absolute value of X
DERIV(X)     first time derivative of X (time-history only)
INTEG(X)     integral of X (time-history only).
BIT10(X)     Boolean AND with the UH60 tail-rotor bit
```

Univariate Table Look-up

Univariate table look-up is also available in TRENDS. The table must be entered as a number of x,y pairs in your user-defined functions file, FUNCTIONS.A/C. If the table is called VGAIN, for example, VGAIN(X) may be used in any mathematical expression as an operand, where the argument X may also be an expression. Linear interpolation is used between table-points when X lies within the table's independent-variable bounds. When X lies outside the bounds, the end-point y-value is returned.

Valid Examples

```
               (Entries in user's FUNCTIONS.703)
AVGTORK = (M143 + M107) / 2                                ! named formula
DATA VGAIN (0,23, 45,62.5, 100,83.7, 200,0, 300,-5)   ! look-up table

               (In-line responses to plotting "Y-CURVE" prompt)
SHAFT TORQUE=M143                          ! itemcode
STORE RSSTORK = M143^2 + (M107^2)^.5       ! formula
TRIG FN = ATAN(SIN(D186)/COS(D186))-D161   ! library functions
PSEUDOS = RSHP * SQRT(SIGP^3)              ! pseudo-items
TABLE VAL = 1.5 * VGAIN(.67*P002)          ! table look-up
FN_OF_STORED_VAL = LOG(RSSTORK)            ! stored formula
SCIENT_NOTATION&TIME = M143*1.E-5*TIME     ! E-notation, TIME
AVG_VIB_TORK (FT-LB) = M143.OSC/12         ! non-default measure
CLIMB_RATE (FT/MIN) = DERIV(P342) * 60     ! calculus operations
AVERAGE TORQUE = AVGTORK/12                ! use of named formula
```

In-line responses for MINMAX also apply to SEARCH responses.

APPENDIX C

PERFORMANCE GROUPS, XV-15 AND UH-60

XV-15 TILTROTOR TIME-HISTORY GROUPS

ITEM CODE GROUPS	FILE TYPE	TIME HISTORY FILE DESCRIPTION
AEROELASTIC	*.TIM	FILTERED, FULL CNTR, 26 I.C.s
ARO: AEROELASTIC	*.RAW	UNFILTERED, FULL CNTR, 20 I.C.s
BLADE	*.RAW	UNFILTERED, FULL CNTR, 50 I.C.s
CONVERSION	*.RAW	UNFILTERED, FULL CNTR, 23 I.C.s
HANDLING QUALITIES	*.TIM	FILTERED, FULL CNTR, 108 I.C.s
HARMONICS	*.SPC	UNFILTERED, 1025 PTS/CNTR, ABFMS
MANEUVERS	*.TIM	FILTERED, FULL CNTR, 88 I.C.s
RADAR	*.TIM	FILTERED, FULL CNTR, 19 I.C.s
RAW WING AEROELASTIC	*.RAW	UNFILTERED, FULL CNTR, 9 I.C.s
SPECTRALS	*.SPC	UNFILTERED, 1025 PTS/CNTR, ABFMS
TRANSFER FUNCTION	*.RAW	UNFILTERED, FULL CNTR, 13 I.C.s

XV-15 HANDLING QUALITIES (PERFORMANCE) ITEM CODES

				FILTER	S RATE	IN RATE
1	A005	C.G. VERT VIBR	G'S	.5	5.0	125.5
2	A019	PILOT SEAT VERT VIBR	G'S	.5	5.0	251
3	A020	COPILOT SEAT VERT VIBR	G'S	.5	5.0	251
4	A300	C.G. LAT VIBR	G'S	.5	5.0	125.5
5	A301	C.G. F/A VIBR	G'S	.5	5.0	125.5
6	A302	PILOT SEAT LAT VIBR	G'S	.5	5.0	251
7	A304	COPILOT SEAT LAT VIBR	G'S	.5	5.0	251
8	A352	C.G. VERT VIBR (SERVO)	G'S	3.0	15.7	31.4
9	A380	PILOT SEAT F/A VIBR	G'S	.5	5.0	251
10	D007	ANGLE OF SIDESLIP	DEG	3.0	15.7	31.4
11	D008	ANGLE OF ATTACK	DEG	3.0	15.7	31.4
12	D009	ROLL ATTITUDE - CABIN	DEG	3.0	15.7	125.5
13	D010	PITCH ATTITUDE - CABIN	DEG	3.0	15.7	125.5
14	D011	YAW ATTITUDE - CABIN	DEG	3.0	15.7	125.5
15	D021	F/A STICK POSITION	%	3.0	15.7	125.5
16	D022	LAT STICK POSITION	%	3.0	15.7	125.5
17	D023	POWER LEVER POSITION	%	3.0	15.7	31.4
18	D024	PEDAL POSITION	%	3.0	15.7	125.5

				FILTER	S RATE	IN RATE
19	D025	FFS F/A CYCLIC STICK POSITION	%	3.0	15.7	31.4
20	D026	FFS LAT STICK POSITION	%	3.0	15.7	31.4
21	D027	FFS RUDDER PEDAL POSITION	%	3.0	15.7	31.4
22	D156	RT PYLON HUB SPRING F/A POS	DEG	3.0	15.7	31.4
23	D157	RT PYLON HUB SPRING LAT POS	DEG	3.0	15.7	31.4
24	D158	RT PYLON COLL. ACTUATOR POS	DEG	3.0	15.7	31.4
25	D159	RT PYLON S/PLATE F/A POSITION	DEG	3.0	15.7	125.5
26	D160	RT PYLON S/PLATE LAT POSITION	DEG	3.0	15.7	125.5
27	D161	RT PYLON CONVERSION POSITION	DEG	1.0	5.2	31.4
28	D181	LT PYLON HUB SPRING F/A POS	DEG	3.0	15.7	31.4
29	D182	LT PYLON HUB SPRING LAT POS	DEG	3.0	15.7	31.4
30	D183	LT PYLON COLL. ACTUATOR POS	DEG	3.0	15.7	31.4
31	D184	LT PYLON S/PLATE F/A POSITION	DEG	3.0	15.7	125.5
32	D185	LT PYLON S/PLATE LAT POSITION	DEG	3.0	15.7	125.5
33	D186	LT PYLON CONVERSION POSITION	DEG	1.0	5.2	31.4
34	D281	ELEVATOR POSITION	DEG	3.0	15.7	31.4
35	D284	RUDDER POSITION	DEG	3.0	15.7	31.4
36	D305	RT MAIN LDG GEAR OLEO EXT POS	INCHES	1.0	5.0	125.5

				FILTER	S RATE	IN RATE
37	D306	F/A SCAS ACTUATOR POSITION	INCHES	6.0	31.4	125.5
38	D307	LATERAL SCAS ACTUATOR POSITION	INCHES	6.0	31.4	125.5
39	D308	DIRECTIONAL SCAS ACTUATOR POS	INCHES	6.0	31.4	125.5
40	D309	PILOT FLAP LEVER POSITION	DEG	3.0	15.7	31.4
41	D314	LT MAIN LDG GEAR ACT. POS	INCHES	1.0	5.2	31.4
42	D315	LT MAIN LDG GEAR OLEO EXT POS	INCHES	1.0	5.2	31.4
43	D317	RT MAIN LDG GEAR ACT. POS	INCHES	1.0	5.0	125.5

44	D318	DIFF.CYCLIC WASHOUT ACT. POS	INCHES	3.0	15.7	31.4
45	D327	ALTITUDE - RADAR ALTIMETER	FEET	1.0	5.2	31.4
46	D349	NOSE LDG GEAR OLEO EXT POS	INCHES	1.0	5.0	125.5
47	D509	RT THROTTLE POSITION	DEG	3.0	15.7	31.4
48	D510	LT THROTTLE POSITION	DEG	3.0	15.7	31.4
49	D617	FLAP POSITION	DEG	3.0	15.7	31.4
50	D645	RT WING AILERON POSITION	DEG	3.0	15.7	125.5
51	D646	LT WING AILERON POSITION	DEG	3.0	15.7	125.5
52	D746	RT COLLECTIVE LVDT	%	10.0	31.4	125.5
53	D747	RT FLAPERON LVDT	%	10.0	31.4	125.5
54	D776	S/S LONG STICK COMMAND	INCHES	3.0	15.7	31.4

				FILTER	S RATE	IN RATE
55	D777	S/S PITCH ATTITUDE COMMAND	DEG	3.0	15.7	31.4
56	D779	S/S LAT STICK COMMAND	INCHES	3.0	15.7	31.4
57	D780	S/S ROLL ATTITUDE COMMAND	DEG	3.0	15.7	31.4
58	D792	S/S PEDAL COMMAND	INCHES	3.0	15.7	125.5
59	D799	LT COLLECTIVE LVDT	%	10.0	31.4	125.5
60	D800	LT FLAPERON LVDT	%	10.0	31.4	125.5
61	E069	RPM CMD	%	3.0	15.7	
62	E070	STBY-MA	?	3.0	15.7	
63	E717	PRIMARY GOV SERVO VALVE	MAMPS	3.0	15.7	125.5
64	E718	PRIMARY GOV RPM ERROR	%	10.0	31.4	125.5
65	E719	PRIMARY GOV #1 LVDT	%	3.0	15.7	125.5
66	E720	PRIMARY GOV ACT. VELOCITY	D/SEC	3.0	15.7	125.5
67	E721	PRIMARY GOV COMMAND RPM	%	3.0	15.7	31.4
68	E722	PRIMARY MONITOR COMMAND RPM	%	3.0	15.7	31.4
69	E723	PRIMARY MONITOR RPM ERROR	%	10.0	31.4	125.5
70	E724	STANDBY GOVERNOR RPM ERROR	%	10.0	31.4	31.4
71	E748	RT COLLECTIVE EXCITER SOLENOID	VOLTS	10.0	31.4	31.4
72	E749	RT FLAPERON EXCITER SOLENOID	VOLTS	10.0	31.4	31.4

				FILTER	S RATE	IN RATE
73	E750	LT COLLECTIVE EXCITER SOLENOID	VOLTS	10.0	31.4	31.4
74	E751	LT FLAPERON EXCITER SOLENOID	VOLTS	10.0	31.4	31.4
75	F030	FFS F/A CYCLIC STICK FORCE	LBS	3.0	15.7	31.4
76	F031	FFS LATERAL STICK FORCE	LBS	3.0	15.7	31.4
77	F033	FFS RUDDER PEDAL FORCE	LBS	3.0	15.7	31.4
78	F162	RT F/A CYCLIC ACTUATOR FORCE	LBS	3.0	15.7	251
79	F163	RT LAT STICK ACTUATOR FORCE	LBS	3.0	15.7	251
80	F164	RT COLLECTIVE ACTUATOR FORCE	LBS	3.0	15.7	251
81	F187	LT F/A CYCLIC ACTUATOR FORCE	LBS	3.0	15.7	251
82	F188	LT LAT STICK ACTUATOR FORCE	LBS	3.0	15.7	251
83	F189	LT COLLECTIVE ACTUATOR FORCE	LBS	3.0	15.7	251
84	F330	F/A CYCLIC STICK FORCE	LBS	3.0	15.7	125.5
85	F331	LATERAL STICK FORCE	LBS	3.0	15.7	125.5
86	F333	RT RUDDER PEDAL FORCE	LBS	3.0	15.7	125.5
87	F334	LT RUDDER PEDAL FORCE	LBS	3.0	15.7	125.5
88	F775	S/S LONG FORCE	LBS	3.0	15.7	125.5
89	F778	S/S LAT FORCE	LBS	3.0	15.7	125.5
90	M107	RT ROTOR MAST TORQUE 12	IN LB	3.0	15.7	251

				FILTER	S RATE	IN RATE
91	M143	LT ROTOR MAST TORQUE 12	IN LB	3.0	15.7	251
92	M791	S/S TORQUE	IN LB	3.0	15.7	251
93	P002	AIRSPEED - NOSE BOOM	KNOTS	1.0	5.0	125.5
94	P342	ALTITUDE - NOSE BOOM	FEET	1.0	5.2	31.4
95	R106	ROTOR RPM	%	3.0	31.4	125.5
96	R328	RT ENGINE FUEL FLOW RATE	LB/HR	1.0	5.0	125.5
97	R329	LT ENGINE FUEL FLOW RATE	LB/HR	1.0	5.0	125.5
98	R338	RT ENGINE N2 RPM	%	6.0	31.4	125.5
99	R339	LT ENGINE N2 RPM	%	6.0	31.4	125.5
100	R503	RT ENGINE N1 RPM	%	3.0	15.7	125.5
101	R515	LT ENGINE N1 RPM	%	3.0	15.7	125.5
102	T322	OAT (ROSEMONT)	DEG C	1.0	5.2	31.4
103	V012	ROLL RATE - CABIN (INCOMPLETE)	D/SEC	3.0	15.7	125.5
104	V013	PITCH RATE -CABIN (INCOMPLETE)	D/SEC	3.0	15.7	125.5
105	V014	YAW RATE - CABIN (INCOMPLETE)	D/SEC	3.0	15.7	125.5
106	V015	ROLL RATE - SCAS	D/SEC	3.0	15.7	125.5
107	V016	PITCH RATE - SCAS	D/SEC	3.0	15.7	125.5
108	V017	YAW RATE - SCAS	D/SEC	3.0	15.7	125.5

UH-60 BLACKHAWK INSTRUMENTATION GROUPS

TC	Test Conditions
AP	Aircraft Parameters
RP	Rotor Parameters
VP	Vibration Parameters
EP	Engine Parameters
DP	Derived Parameters

UH-60 BLACKHAWK TEST-CONDITION ITEMS

Seq		Item	Description	Units	Icode	Freq	Rate/Dec
1	T	ALPHA	Angle of attack	Degs	DAA0		32/1
2	T	AXCG	Linear accel CG, longitudinal	G's	DL00	5.	129/4
3	T	AYCG	Linear accel CG, lateral	G's	DL01	5.	129/4
4	T	AYCGSENS	Sensitive lateral acceleration	G's	AF90	5.	129/4
5	T	AZCG	Linear accel CG, normal	G's	DL02	5.	129/4
6	T	BETA	Angle of sideslip	Degs	DSS0		32/1
7		HEADIN	Aircraft Heading @25sps	Degs	DA12		32/1
8	T	HEADING	Aircraft Heading	Degs	DA02	5.	129/4
9	T	LSSX	Raw Airspeed (LASSIE) Long	Kts	VX03	1.	32/6
10	T	LSSY	Raw Airspeed (LASSIE) Lateral	Kts	VY03	1.	32/6
11	T	LSSZ	Raw Airspeed (LASSIE) Vertical	Ft/Min	VZ03	1.	32/6
12	T	PAICB	Boom altitude	inHg	H001	1.	32/6
13	T	PAICS	Ship's altitude	inHg	H002	1.	32/6
14		PITCHAT	Pitch attitude @ 25 sps	Degs	DA10		32/6
15	T	PITCHATT	Attitude, pitch angle	Degs	DA00	5.	129/4
16	T	PTCHACC	Pitch acceleration	Deg/s2	DAC0	2.	129/8
17	T	PTCHRATE	Angular rate, pitch	Deg/s	DR00	5.	129/4
18	T	QCICB	Boom airspeed	inHg	V001	1.	32/6
19	T	QCICS	Ship's airspeed	inHg	V002	1.	32/6
20	T	RADALT	Altitude (Radar Range)	Feet	H003	1.	32/6
21		RECORD	Record number	Index			32/6
22	T	ROLLACC	Roll acceleration	Deg/s2	DAC1	2.	129/8
23		ROLLAT	Roll attitude @ 25 sps	Degs	DA11		129/8
24	T	ROLLATT	Attitude, roll angle	Degs	DA01	5.	129/4
25	T	ROLLRATE	Angular rate, roll	Deg/s	DR01	5.	129/4
26	T	TTIC	OAT Outside Air Temperature	Deg C	T100	1.	32/6
27	T	YAWACC	Yaw acceleration	Deg/s2	DAC2	2.	129/8
28		YAWATT	Alternate for heading	Deg	DA22		129/8
29	T	YAWRATE	Angular rate, yaw	Deg/s	DR02	5.	129/4

APPENDIX D

FILE STRUCTURE CONSIDERATIONS

One must consider many factors when choosing a file structure for a database. Some of these factors are:

1. How will the information from the database be used?
2. How much data must be archived?
3. What are the procedural limitations and constraints?

The basic keys for databased numerical (e.g., Min/Max or time-history) records are (1) item name (itemcode, mnemonic, etc.) and (2) counter (test-point identifier). That is, given the data item and test-point index, the data record should be identified for retrieval. Of course, We are assuming that the database is given and that item and counter are unique within that database. Such attributes as data type or repeated records will be ignored for now. There are many file and record structures one could propose for numerical records:

Option 1: All of the item/counter data might be in one file keyed either by a composite item/counter code or by primary (e.g., item) and secondary (e.g., counter) keys

Option 2: All of the data in the base for one item might be in one file (one file per item) and be keyed by counter

Option 3: All of the data in the base for one counter might be in one file (one file per counter) and be keyed by item

Option 4: Each item/counter pair might be a separate file (name code specifies item and counter, no keying necessary)

The following FORTRAN code shows how Option 2 is implemented in TRENDS.

```
      OPEN (UNIT=2, NAME=FILENAME, STATUS='OLD',
     1     ORGANIZATION='INDEXED', ACCESS='KEYED',
     2     KEY=(1:4:INTEGER), READONLY, SHARED,
     3     RECORDTYPE='VARIABLE', FORM='UNFORMATTED')

      JCTR = 14502          ! Counter number

      READ (2,KEY=JCTR,ERR=350) ICTR, SCALE, BIAS, START, SAMPSEC,
     1      NPTS, (I2DAT(I),I=1,NPTS)

      DO 200 I=1,NPTS
200      EU(I) = SCALE * FLOAT(I2DAT(I)) + BIAS
```

This example reads time-history data in integer counts into an array, I2DAT, then converts the data into an array, EU, in floating-point engineering units for plotting or math operations. Notice that the key is the counter number, an integer. For Option 3, the record key would be the itemcode, a character variable. The choice among the many options is made by considering the likely use of the data, the ease of programming the access, database management requirements (e.g., deleting, copying the stored data), resultant file size, traceability through modularity, etc.

Option 2 was originally selected for TRENDS for storing both Min/Max and time-history data. Reconsideration of the usage requirements and constraints of the system has led to a decision to change time-history file structures to Option 3 in the future. Option 1 was ruled out because it implies one gigantic file for each data type (or worse, for ALL data types, such as Min/Max-per-counter, filtered time-histories, etc.). This file would be too unwieldy for database management operations where special techniques are required to avoid locking up files while filling and where smaller files are better procedurally for accommodating such problems as computer crashes. Option 2 implies a fairly fixed number of files (one for each item), each of which grows as new counters are added to the database. Option 3 implies a fairly fixed number of records in each file (file size depending on test-point duration and sampling rate), where the number of files grows as new counters are added to the database. Option 4 would result in a ridiculous proliferation of files (the number being the product of the number of items and the number of counters for each data type) and in too much opening and closing of files during access. Time-frame format storage is not even considered, because it takes too much work and time to extract the information to be displayed or searched.

The multiflight requirement is currently impacting the time-history data format as the system goes from 5 to 80 gigabytes via the use of a laser-disk jukebox. The original file format (Option 2) was multi-flight-oriented by allowing each parameter to have a single file and by keying the records in it to the test-point numbers. New higher data sampling rates (for multiple flights) are forcing NASA to go to a multidisk storage system to handle the high volume of data, where the time-history data for one parameter might well span more than one disk if kept in the original format. Option 3, which has one file per test point (the file data for all parameters for that test point) is better for this situation because each disk could be made to hold an integral number of test points. This format may not be as efficient in execution as that of Option 2, but has logistical advantages because of the laser disk medium and the need to migrate data.

A version of the Option 3 format has been adopted at Ames to serve as the standard database time-history format serving both DATAMAP and TRENDS. This format is called the DAT-TH format.

The format requirements for Min/Max data are different from those for time-history data. Let us consider the advantages and disadvantages of using Option 2 (one file per item, keyed by counter) for Min/Max operations. The two most frequent uses of Min/Max data in TRENDS are (1) searching for conditions on a few items across a possibly large range of counters, and (2) displaying one item's Min/Max statistics versus another's for a range of counters. To search one item's values across the entire database (perhaps thousands of counters) requires opening only one file. To display one item's values against another's for a large number of counters requires opening and reading only two files. If the files were counter-named and item-keyed (Option 3), a new file would have to be opened, read, and closed for each different counter in the specified data region. A disadvantage of Option 2 is seen in block-prints (snapshots displaying multiple items or expressions involving multiple items) of many items together for a few counters. Another difficulty comes in deleting (or migrating) one counter or flight or segment from the database. Each item-file must then be specified, opened, treated, and closed separately. On the other hand, the item-file for a completely dead or faulty sensor can simply be deleted and full database information for a single item can be copied with VMS system commands.

APPENDIX E
HIERARCHY CHART FOR TRENDS

The subroutines are :

AMPSET	AMPSEC	ASKFILE	AVECYC	AZIMUTH
BANK	BLDFORMAT	BLOCKOUT	BUZZER	CAST
CHANGE	CHANGE_SETUP	CHEKFUNC	CKDATNAM	COMPARE
COMPHDR	CONTENTS	CREATION	CRSHAR	CUBFIT
CURVEFIT	CVFILT	CVGEN	CVTUPC	CYAVST
CYCLAVG	DCSREAD	DCSSAVE	DERIVED	DISPLT
DRWCRV	ECR	EDITPP	ENABLE	EQUATION
EQUATIONF	EVALUATE	EXPOSE	EXPOSEF	FFT
FILDAT	FILTER	FINDTHC	FLAGCHEK	FLT2CTR
FLTLOG	FMNMAX	FMTRATE	FOPEN	FORPIC
FRFILL	FRMAT	FUNC	FUNKSHN	GENHELP
GETERM	GETFLTS	GETHMC	GETNPLOT	GETNPLOTF
GICSUB	GRPLIST	HANDLER	HELP	HELPER
HELPHDCPY	INDATA	INFILES	INFUNCS	INREAD
INVERT	ITEMCODE	ITEMDEFS	ITEMVU	LABELS
LEGENDS	LIBDD	LOADSUH	LOADSXV	LOGOUT
MENU	MINMAX	MPLOTS	MSKFILE	MULTIPLT
NICER	NORMALIZE	NOTCHF	ONELINER	OPENCNTRF
OPENFILE	OPSHNS	PARSER	PERFPLOT	PICTR
PLTHDR	PLTLABLS	PLTSET	PLTSETUP	PLTSETUPF
PREPLOT	PREPLOTF	QWIKPLOT	READUPC	REDNOSE
RFFT	RNUMB	RNUMBA	RNUMBR	ROTCOR
ROTDEG	ROTPUL	SCLAREP	SCRO_MENU	SEARCHUH
SEARCHXV	SECYAX	SETACSN	SETBASE	SETERM
SETSCL	SETUP1	SETUP2	SHIFTY	SHOCAL
SHOFLT	SHOKEY	SHOMNMX	SHONARTV	SHOSUM
SHOWDCS	SHOWFILE	SHOWFORM	SHOWTH	SHOWVAR
SIMSIGNAL	SIMSPEC	SMG_MENU	SORTI4	SORTX
SPECTRAL	STRINGS	SUBMENU	TEKVT	TERMTYPE
THFOPEN	THITEMS	THPRNT	TIMEHIST	TIMEOUT
TREND98	UPDSTATS	USERFILES	WHATFILES	WHATSAVE
WORDSCAN	XAXIS	XKMSCL	YAXIS	YKMSCL

In the following hierarchy chart, "*" means "not in the list of subroutines" and "^line n" means "lower branches already expanded above, line n."

```
       Level Level Level Level Level .........
         0     1     2     3     4   .........

  1    TREND98                               (TRENDS Main Program)
  2    >     CHANNEL      *
  3    >     ENABLE
  4    >     >     LIB$STOP     *
  6    >     LOGOUT
  7    >     >     DATE         *
  8    >     >     TIME         *
 10    >     SETACSN
```

```
11  >    >    READUPC
12  >    >    INFUNCS
13  >    >    >    CVTUPC
14  >    >    >    PARSER
16  >    >    SETBASE
17  >    >    >    STRINGS
20  >    SET_TERM_HCPY *
21  >    ECR
22  >    INFUNCS        ^line  12
23  >    INFILES
24  >    MENU                              (TRENDS Menu Prompter)
25  >    >    SCRO_MENU
26  >    >    >    READUPC
28  >    >    SMG_MENU
29  >    >    >    OPSHNS
30  >    >    >    CVTUPC
31  >    >    >    CHANGE_SETUP
32  >    >    >    >    SUBMENU
33  >    >    >    >    SETACSN        ^line  10
36  >    >    GENHELP
37  >    >    >    READUPC
38  >    >    >    HELP
39  >    >    >    HELPER
40  >    >    >    >    READUPC
41  >    >    >    >    LOGOUT         ^line   6
42  >    >    >    >    ITEMDEFS
43  >    >    >    >    >    LOGOUT         ^line   6
44  >    >    >    >    >    FOPEN
45  >    >    >    >    >    READUPC
46  >    >    >    >    >    CVTUPC
47  >    >    >    >    >    FMTRATE
48  >    >    >    >    >    HELP
50  >    >    >    >    SHOWFORM
51  >    >    >    >    WHATSAVE
52  >    >    >    >    >    READUPC
54  >    >    >    >    HELPHDCPY
55  >    >    >    >    HELP
56  >    >    >    >    SHOWTH
57  >    >    >    >    FINDTHC                          (FIND Menu Item)
58  >    >    >    >    >    LOGOUT         ^line   6
59  >    >    >    >    >    READUPC
60  >    >    >    >    >    CONTENTS
61  >    >    >    >    >    >    FOPEN
63  >    >    >    >    >    THITEMS
64  >    >    >    >    >    >    READUPC
65  >    >    >    >    >    >    REDNOSE         (Number entry parser)
66  >    >    >    >    >    >    >    READUPC
67  >    >    >    >    >    >    >    CHANGE
68  >    >    >    >    >    >    >    >    PARSER
69  >    >    >    >    >    >    >    >    SETACSN        ^line  10
70  >    >    >    >    >    >    >    >    GETERM
71  >    >    >    >    >    >    >    >    >    READUPC
```

```
                                                                GETHCPY          *
73  >    >    >    >    >    >    >    >    GETHCPY          *
74  >    >    >    >    >    >    >    >    THPRNT
75  >    >    >    >    >    >    >    >    >    HELP
77  >    >    >    >    >    >    >    >    EDITPP
78  >    >    >    >    >    >    >    >    >    WHATFILES
79  >    >    >    >    >    >    >    >    >    READUPC
80  >    >    >    >    >    >    >    >    >    PARSER
82  >    >    >    >    >    >    >    >    READUPC
83  >    >    >    >    >    >    >    >    RNUMBR
84  >    >    >    >    >    >    >    >    >    RNUMB
87  >    >    >    >    >    >    >    DCSREAD
88  >    >    >    >    >    >    >    >    ASKFILE
89  >    >    >    >    >    >    >    >    >    READUPC
93  >    >    >    >    >    >    SORTI4
94  >    >    >    >    >    >    DCSSAVE
95  >    >    >    >    >    >    >    READUPC
96  >    >    >    >    >    >    >    ASKFILE         ^line  88
97  >    >    >    >    >    >    >    CREATION
98  >    >    >    >    >    >    >    >    DATE             *
99  >    >    >    >    >    >    >    >    TIME             *
103 >    >    >    >    >    STRINGS
104 >    >    >    >    >    FOPEN
105 >    >    >    >    >    REDNOSE         ^line  65
106 >    >    >    >    >    THFOPEN
107 >    >    >    >    >    SHOWTH
108 >    >    >    >    >    SHOWDCS
109 >    >    >    >    >    FLT2CTR
110 >    >    >    >    >    >    SORTI4
112 >    >    >    >    >    OPENCNTRF
113 >    >    >    >    >    >    FOPEN
115 >    >    >    >    >    INREAD
116 >    >    >    >    >    SORTI4
117 >    >    >    >    >    DCSSAVE         ^line  94
119 >    >    >    >    SHOWDCS
120 >    >    >    >    SHOMNMX
122 >    >    >    LOGOUT          ^line   6
123 >    >    >    HELPHDCPY
125 >    >    SETACSN         ^line  10
126 >    >    GETERM          ^line  70
127 >    >    GETHCPY         *
128 >    >    HELP                                   (HELP Menu Item)
129 >    >    LIBDO
130 >    >    >    LIB$SIGNAL      *
133 >    SHOSUM                                      (DATABASE Menu Item)
134 >    >    SHOWTH
135 >    >    SHOMNMX
136 >    >    SHONARTV
137 >    >    READUPC
138 >    >    LOGOUT          ^line   6
139 >    >    ONELINER
141 >    SHOFLT                                      (FLIGHTS Menu Item)
142 >    >    LOGOUT          ^line   6
```

```
143  >    >    READUPC
144  >    >    DCSREAD        ^line  87
145  >    >    FOPEN
146  >    >    PARSER
147  >    >    SHONARTV
148  >    >    SORTI4
149  >    >    DCSSAVF        *
151  >    FLTLOG                               (LOGSCAN Menu Item)
152  >    >    LOGOUT         ^line   6
153  >    >    READUPC
155  >    SHOKEY                               (KEYS Menu Item)
156  >    >    LOGOUT         ^line   6
157  >    >    FOPEN
158  >    >    REDNOSE        ^line  65
159  >    >    SHOWDCS
160  >    >    FLT2CTR        ^line 109
161  >    >    FLAGCHEK
163  >    ITEMVU                               (VIEW Menu Item)
164  >    >    LOGOUT         ^line   6
165  >    >    FOPEN
166  >    >    REDNOSE        ^line  65
167  >    >    SHOMNMX
168  >    >    SHOWDCS
169  >    >    FLT2CTR        ^line 109
170  >    >    READUPC
171  >    >    HELP
172  >    >    DEFITEMSUH     *
173  >    >    FLAGCHEK
175  >    BLOCKOUT                             (CPRINT Menu Item)
176  >    >    LOGOUT         ^line   6
177  >    >    READUPC
178  >    >    BLDFORMAT
179  >    >    >    READUPC
181  >    >    STRINGS
182  >    >    REDNOSE        ^line  65
183  >    >    FLT2CTR        ^line 109
184  >    >    EVALUATE
185  >    >    >    EQUATION
186  >    >    >    >    FOPEN
188  >    >    >    FUNC
191  >    SEARCHXV                             (SEARCH Menu Item, XV-15)
192  >    >    LOGOUT         ^line   6
193  >    >    DATE           *
194  >    >    TIME           *
195  >    >    FOPEN
196  >    >    READUPC
197  >    >    MSKFILE
198  >    >    >    READUPC
200  >    >    HELP
201  >    >    EQUATION       ^line 185
202  >    >    RNUMBA
203  >    >    >    RNUMB
```

```
205   >    >    CREATION      ^line  97
206   >    >    REDNOSE       ^line  65
207   >    >    SHOWDCS
208   >    >    FUNC
209   >    >    FRMAT
210   >    >    UPDSTATS
211   >    >    SORT14
212   >    >    DCSSAVE       ^line  94
214   >    SEARCHUH                                (SEARCH Menu Item, not XV-15)
215   >    >    FOPEN
216   >    >    LOGOUT        ^line   6
217   >    >    DATE          *
218   >    >    TIME          *
219   >    >    READUPC
220   >    >    MSKFILE       ^line 197
221   >    >    HELP
222   >    >    RNUMBA        ^line 202
223   >    >    EQUATION      ^line 185
224   >    >    CREATION      ^line  97
225   >    >    REDNOSE       ^line  65
226   >    >    SHOWDCS
227   >    >    FLAGCHEK
228   >    >    FUNC
229   >    >    FRMAT
230   >    >    UPDSTATS
231   >    >    SORT14
232   >    >    DCSSAVE       ^line  94
234   >    WORDSCAN                                (WORDSCAN Menu Item)
235   >    >    LOGOUT        ^line   6
236   >    >    READUPC
237   >    >    HELP
238   >    >    REDNOSE       ^line  65
239   >    >    HANDLER
240   >    >    >     THITEMS       ^line  63
242   >    >    SHOWDCS
243   >    >    FLT2CTR       ^line 109
244   >    >    TIMEOUT
245   >    >    SORT14
246   >    >    DCSSAVE       ^line  94
248   >    SHOCAL                                  (CALIBS Menu Item)
249   >    >    LOGOUT        ^line   6
250   >    >    READUPC
251   >    >    HELP
252   >    >    REDNOSE       ^line  65
253   >    >    GETFLTS
255   >    MINMAX                                  (MINMAX Menu Item)
256   >    >    HELP
257   >    >    TERMTYPE
258   >    >    >     READUPC
260   >    >    READUPC
261   >    >    EDITPP        ^line  77
262   >    >    HELPER        ^line  39
```

```
263  >    >    LOGOUT        ^line   6
264  >    >    EQUATION      ^line 185
265  >    >    PARSER
266  >    >    SCLAREP
267  >    >    SIMSPEC
268  >    >    >    READUPC
270  >    >    REDNOSE       ^line  65
271  >    >    FLT2CTR       ^line 109
272  >    >    FLAGCHEK
273  >    >    LABELS
274  >    >    >    PLTHDR
275  >    >    >    >    FOPEN
276  >    >    >    >    COMPHDR
277  >    >    >    >    >    FOPEN
279  >    >    >    >    SETBASE       ^line  16
281  >    >    >    XAXIS
282  >    >    >    FOPEN
283  >    >    >    YAXIS
284  >    >    >    >    LEGENDS
287  >    >    FUNC
288  >    >    SORTX
289  >    >    PREPLOT
290  >    >    >    SPECTRAL
291  >    >    >    >    AMPSET
292  >    >    >    >    >    AMPSPC
293  >    >    >    >    >    >    RFFT
294  >    >    >    >    >    >    >    FFT
298  >    >    >    >    GETMMC
300  >    >    >    CYCLAVG
301  >    >    >    >    AZIMUTH
302  >    >    >    >    >    OPENFILE
303  >    >    >    >    >    >    THFOPEN
305  >    >    >    >    >    INDATA
306  >    >    >    >    >    >    OPENCNTRF      ^line 112
307  >    >    >    >    >    >    INREAD
309  >    >    >    >    >    ROTPUL
310  >    >    >    >    >    ROTDEG
311  >    >    >    >    >    ROTCOR
313  >    >    >    >    CYAVST
314  >    >    >    >    >    AVECYC
315  >    >    >    >    >    >    CUBFIT
319  >    >    >    SETSCL
320  >    >    >    CURVEFIT
321  >    >    >    >    INVERT
324  >    >    PLTSET
325  >    >    DISPLT
326  >    >    >    FORPIC
327  >    >    >    >    RNUMBR        ^line  83
328  >    >    >    >    FRFILL
329  >    >    >    >    >    SORTX
331  >    >    >    >    FMNMAX
332  >    >    >    >    PICTR
```

```
334  >    >    >    BUZZER
335  >    >    >    READUPC
336  >    >    >    SETERM
337  >    >    >    >       REGIS          *
338  >    >    >    >       REGNOM         *
339  >    >    >    >       TK4052         *
340  >    >    >    >       TK4014         *
341  >    >    >    >       TK4010         *
342  >    >    >    >       TKNOM          *
343  >    >    >    >       TEKVT
344  >    >    >    >       IOMGR          *
345  >    >    >    >       HP2623         *
346  >    >    >    >       H26NOM         *
347  >    >    >    >       DEVT           *
348  >    >    >    >       DIP            *
349  >    >    >    >       DICOMD         *
350  >    >    >    >       COMPRS         *
351  >    >    >    >       SETDEV         *
353  >    >    >    SETUP1
354  >    >    >    >       RESET          *
355  >    >    >    >       NOBRDR         *
356  >    >    >    >       PAGE           *
357  >    >    >    >       HWROT          *
358  >    >    >    >       LEGLIN         *
359  >    >    >    >       GRACE          *
360  >    >    >    >       INTAXS         *
361  >    >    >    >       XREVTK         *
362  >    >    >    >       YREVTK         *
363  >    >    >    >       XTICKS         *
364  >    >    >    >       YTICKS         *
366  >    >    >    FRFILL            ^line 328
367  >    >    >    SETUP2
368  >    >    >    >       PHYSOR         *
369  >    >    >    >       AREA2D         *
370  >    >    >    >       HEIGHT         *
371  >    >    >    >       LINESP         *
372  >    >    >    >       LINES          *
373  >    >    >    >       RESET          *
375  >    >    >    SEGREG            *
376  >    >    >    PLTLABLS
377  >    >    >    >       HEADIN         *
378  >    >    >    >       ENDGR          *
379  >    >    >    >       PHYSOR         *
380  >    >    >    >       AREA2D         *
381  >    >    >    >       HEIGHT         *
382  >    >    >    >       XNAME          *
383  >    >    >    >       XNONUM         *
384  >    >    >    >       XTICKS         *
385  >    >    >    >       YNAME          *
386  >    >    >    >       RESET          *
387  >    >    >    >       YNONUM         *
388  >    >    >    >       GRAF           *
```

```
389  >    >    >    >    NUMODE          *
390  >    >    >    >    XKMSCL
391  >    >    >    >    >    HEIGHT          *
392  >    >    >    >    >    MESSAG          *
393  >    >    >    >    >    RESET           *
395  >    >    >    >    YKMSCL
396  >    >    >    >    >    HEIGHT          *
397  >    >    >    >    >    ANGLE           *
398  >    >    >    >    >    MESSAG          *
399  >    >    >    >    >    RESET           *
402  >    >    >    DRWCRV
403  >    >    >    >    SCLPIC          *
404  >    >    >    >    GRID            *
405  >    >    >    >    DOT             *
406  >    >    >    >    RESET           *
407  >    >    >    >    MARKER          *
408  >    >    >    >    CHNDSH          *
409  >    >    >    >    CHNDOT          *
410  >    >    >    >    DASH            *
411  >    >    >    >    PSMTH           *
412  >    >    >    >    CURVE           *
413  >    >    >    >    VECTOR          *
415  >    >    >    SECYAX
416  >    >    >    >    HEIGHT          *
417  >    >    >    >    YNONUM          *
418  >    >    >    >    YGRAXS          *
419  >    >    >    >    YKMSCL          ^line 395
420  >    >    >    >    ANGLE           *
421  >    >    >    >    MESSAG          *
422  >    >    >    >    RESET           *
423  >    >    >    >    DASH            *
424  >    >    >    >    CHNDSH          *
425  >    >    >    >    MARKER          *
426  >    >    >    >    CURVE           *
427  >    >    >    >    INTNO           *
428  >    >    >    >    REALNO          *
429  >    >    >    >    LEGEND          *
431  >    >    >    SEGEND          *
432  >    >    >    CHVIS           *
433  >    >    >    CRSHAR
434  >    >    >    >    CRDSYS          *
435  >    >    >    >    INILOC          *
436  >    >    >    >    ECHLOC          *
437  >    >    >    >    HEIGHT          *
438  >    >    >    >    SEGBEG          *
439  >    >    >    >    CHVIS           *
440  >    >    >    >    REQLOC          *
441  >    >    >    >    SEGDEL          *
442  >    >    >    >    MESSAG          *
443  >    >    >    >    REALNO          *
444  >    >    >    >    SEGEND          *
446  >    >    >    ENDGR           *
```

```
447   >     >     >     DEFERL          *
448   >     >     >     SEGDEL          *
449   >     >     >     ENDPL           *
450   >     >     >     TEKVT
                                                    (QWIKPLOT Menu Item)
453   >     QWIKPLOT
454   >     >     TERMTYPE       ^line 257
455   >     >     READUPC
456   >     >     HELP
457   >     >     FOPEN
458   >     >     REDNOSE        ^line  65
459   >     >     SHOWTH
460   >     >     SHOWDCS
461   >     >     OPENFILE       ^line 302
462   >     >     OPENCNTRF      ^line 112
463   >     >     PARSER
464   >     >     FLAGCHEK
465   >     >     INREAD
466   >     >     LABELS         ^line 273
467   >     >     FUNC
468   >     >     PREPLOT        ^line 289
469   >     >     PLTSET
470   >     >     FORPIC         ^line 326
471   >     >     DISPLT         ^line 325
                                                    (HARMONIC Menu Item, in MINMAX)
473   >     HARMONIC        *
                                                    (MULTIPLT Menu Item)
474   >     MULTIPLT
475   >     >     HELP
476   >     >     TERMTYPE       ^line 257
477   >     >     READUPC
478   >     >     EQUATION       ^line 185
479   >     >     PARSER
480   >     >     REDNOSE        ^line  65
481   >     >     SHOWDCS
482   >     >     FLT2CTR        ^line 109
483   >     >     FLAGCHEK
484   >     >     LABELS         ^line 273
485   >     >     FUNC
486   >     >     SORTX
487   >     >     PREPLOT        ^line 289
488   >     >     PLTSET
489   >     >     RNUMBA         ^line 202
490   >     >     DISPLT         ^line 325
                                                    (NORMALIZ Menu Item)
492   >     NORMALIZE
493   >     >     TERMTYPE       ^line 257
494   >     >     PLTSETUPF
495   >     >     >     INFUNCS         ^line  12
496   >     >     >     READUPC
497   >     >     >     HELP
498   >     >     >     WHATFILES
499   >     >     >     THPRNT          ^line  74
500   >     >     >     PARSER
501   >     >     >     EQUATIONF
502   >     >     >     >     FOPEN
```

```
504  >    >    >    FOPEN
505  >    >    >    BANK
507  >    >    GETNPLOTF
508  >    >    >    SIMSIGNAL
509  >    >    >    >    RNUMBR          ^line  83
510  >    >    >    >    CVGEN
511  >    >    >    >    CVFILT
513  >    >    >    REDNOSE        ^line  65
514  >    >    >    SHOWTH
515  >    >    >    SHOWDCS
516  >    >    >    OPENCNTRF      ^line 112
517  >    >    >    BANK
518  >    >    >    OPENFILE       ^line 302
519  >    >    >    FOPEN
520  >    >    >    INREAD
521  >    >    >    EXPOSEF
522  >    >    >    >    FUNC
523  >    >    >    >    FILTER
524  >    >    >    >    >    CVGEN
525  >    >    >    >    >    CVFILT
527  >    >    >    >    NOTCHF
528  >    >    >    >    >    INVERT
530  >    >    >    >    SORTX
531  >    >    >    >    PREPLOTF
532  >    >    >    >    >    SPECTRAL       ^line 290
533  >    >    >    >    >    SETSCL
534  >    >    >    >    >    CURVEFIT       ^line 320
536  >    >    >    >    LABELS         ^line 273
537  >    >    >    >    PLTSET
538  >    >    >    >    FORPIC         ^line 326
539  >    >    >    >    DISPLT         ^line 325
543  >    ITEMDEFS      ^line  42                 (ITEMDEFS Menu Item)
544  >    SPECTRA       *               (SPECTRAL Menu Item, in TIMEHIST)
545  >    COMPARE                                  (COMPARE Menu Item)
546  >    >    HELP
547  >    >    READUPC
548  >    >    COMPARETH      *
549  >    >    SETACSN        ^line  10
551  >    LOADSXV                          (LOADS Menu Item, for XV-15)
552  >    >    LOGOUT         ^line   6
553  >    >    FOPEN
554  >    >    READUPC
555  >    >    HELP
556  >    >    HELPHDCPY
557  >    >    THFOPEN
558  >    >    REDNOSE        ^line  65
559  >    >    NICER
560  >    >    PICTR
562  >    LOADSUH                          (LOADS Menu Item, not XV-15)
563  >    >    LOGOUT         ^line   6
564  >    >    FOPEN
565  >    >    READUPC
```

```
566  >      >       HELP
567  >      >       HELPHDCPY
568  >      >       THFOPEN
569  >      >       REDNOSE         ^line  65
570  >      >       FLAGCHEK
571  >      >       MICER
572  >      >       FICTR
574  >     TIMEHIST                                          (TIMEHIST,SPECTRAL Menu Items)
575  >      >       HELP
576  >      >       TERMTYPE        ^line 257
577  >      >       PLTSETUP
578  >      >       >       READUPC
579  >      >       >       EDITPP          ^line  77
580  >      >       >       UNSAVE          *
581  >      >       >       WHATSAVE        ^line  51
582  >      >       >       HELP
583  >      >       >       HELPER          ^line  39
584  >      >       >       THPRNT          ^line  74
585  >      >       >       PARSER
586  >      >       >       EQUATION        ^line 185
587  >      >       >       FOPEN
588  >      >       >       BANK
589  >      >       >       SCLAREP
591  >      >       GETNPLOT
592  >      >       >       SIMSIGNAL       ^line 508
593  >      >       >       REDNOSE         ^line  65
594  >      >       >       HELPER          ^line  39
595  >      >       >       FLAGCHEK
596  >      >       >       SETBASE         ^line  16
597  >      >       >       BANK
598  >      >       >       OPENCNTRF       ^line 112
599  >      >       >       OPENFILE        ^line 302
600  >      >       >       INREAD
601  >      >       >       EXPOSE
602  >      >       >       >       FUNC
603  >      >       >       >       FILTER          ^line 523
604  >      >       >       >       NOTCHF          ^line 527
605  >      >       >       >       SORTX
606  >      >       >       >       SHIFTY
607  >      >       >       >       PREPLOT         ^line 289
608  >      >       >       >       BANK
609  >      >       >       >       LABELS          ^line 273
610  >      >       >       >       PLTSET
611  >      >       >       >       DISPLT          ^line 325
615  >     USERFILES                                          (FILES Menu Item)
616  >      >       DATE            *
617  >      >       HELP
618  >      >       READUPC
619  >      >       CKDATNAM
621  >     FINDTEC         ^line  57                          (FIND Menu Item)
622  >     FUNKSEN                                            (FUNCTION Menu Item)
623  >      >       CVTUPC
```

```
624  >    >    EQUATION      ^line 185
625  >    >    READUPC
626  >    >    CHEKFUNC
627  >    >    >    INFUNCS       ^line  12
628  >    >    >    CVTUPC
629  >    >    >    EQUATION      ^line 185
632  >    REVDATA        *
633  >    PERFPLOT                                    (PERFPLOT Menu Item)
634  >    >    READUPC
635  >    >    SHOWVAR
636  >    >    >    SHOWFILE
637  >    >    >    STRINGS
638  >    >    >    REDNOSE       ^line  65
639  >    >    >    READUPC
640  >    >    >    ITEMCODE
641  >    >    >    >    FOPEN
644  >    >    TERMTYPE      ^line 257
645  >    >    SHOWFILE
646  >    >    SHOWVAR1      *
647  >    >    REDNOSE       ^line  65
648  >    >    SHOWTH
649  >    >    SHOWDCS
650  >    >    OPENCNTRF     ^line 112
651  >    >    FILDAT
652  >    >    >    FLAGCHEK
653  >    >    >    OPENCNTRF     ^line 112
654  >    >    >    INREAD
655  >    >    >    THFOPEN
657  >    >    BUZZER
658  >    >    SETERM        ^line 336
659  >    >    MPLOTS
660  >    >    >    PLTHDR        ^line 274
661  >    >    >    RESET         *
662  >    >    >    NOBRDR        *
663  >    >    >    PAGE          *
664  >    >    >    HWROT         *
665  >    >    >    HEIGHT        *
666  >    >    >    LEGLIN        *
667  >    >    >    INTAXS        *
668  >    >    >    XREVTK        *
669  >    >    >    YREVTK        *
670  >    >    >    XTICKS        *
671  >    >    >    YTICKS        *
672  >    >    >    PHYSOR        *
673  >    >    >    AREA2D        *
674  >    >    >    HEADIN        *
675  >    >    >    ENDGR         *
676  >    >    >    FILDAT        ^line 651
677  >    >    >    MESSAG        *
678  >    >    >    ITEMCODE      ^line 640
679  >    >    >    SETSCL
680  >    >    >    XNAME         *
```

```
681  >     >     >     YNAME        *
682  >     >     >     GRAF         *
683  >     >     >     DOT          *
684  >     >     >     GRID         *
685  >     >     >     CURVE        *
686  >     >     >     ENDPL        *
687  >     >     >     TEKVT
690  >     GRPLIST                                      (GROUPS Menu Item)
691  >     >     FOPEN
692  >     >     LOGOUT          ^line    6
693  >     >     READUPC
694  >     >     GICSUB
696  >     DERIVED                                      (DERIVED Menu Item)
697  >     >     FOPEN
698  >     >     LOGOUT          ^line    6
699  >     >     READUPC
700  >     >     HELP
701  >     >     HELPHDCPY
703  >     DONEPL          *
704  >     LIBDO           ^line 129
```

APPENDIX F

DATABASE VECTORING

Current NASA Procedures for Data Access: Commands are currently constructed in TRENDS, running under VMS, to allow the user to access any database which can be stored on one of five different disk drives. In the following example, a path vector is constructed, then used to retrieve file data off a magnetic disk farm.

```
$ ASSIGN NEP1:[TRENDS]DBASE.RUN  FOR098
$ RUN NEP1:[TRENDS]XVTRENDS
```

```
Note: In the above assignment NEP1      =   disk name (NEPTUNE)
                              TRENDS    =   main TRENDS directory (DB Op. Sys)
                              DBASE.RUN =   database pointer file
                              FOR098    =   logical unit 98
                              XVTRENDS  =   executable TRENDS program
```

Note: When the user enters into TRENDS, he is requested to select his database. e.g. 748, 703, 702, etc. (various rotorcraft databases). His selection determines the path (drive name and directory) to the database as follows:

```
1. $ Read unit 98                    ! find the path to sections of databases
                                     ! Note: Different sections (files) of a
                                     ! database can be on different disks for
                                     ! the same rotorcraft.  Contents of file
                                     ! 98 are identified by key words, e.g.
                                     ! DRIVER, DOC, DATA, etc.
2. $ Construct Filename              ! Access data by concatenating path
                                     ! from unit 98 with dataset file names

 e.g.
   FILENAME = 'NEP3:[DB703]'//'C703_13258.225'
              (from unit 098)    (file name)

3. $ Open Filename                   ! e.g. OPEN (UNIT=1, NAME=FILENAME, -
                                     ! STATUS='OLD')
4. $ Read Data                       ! e.g. READ (1) data

    - - - - - - - - > Contents of file 98:  (abbreviated example)

! Rotorcraft #748 -------------------------------------------
        748%DRIVER   %NEP1:[TRENDS.UH60]    ! Unique code for rotorcraft
        748%DOC      %NEP1:[TRENDS]         ! Generic help files
        748%DATA     %NEP2:[DB748]          ! rotorcraft data

! Rotorcraft #703 -------------------------------------------
        703%DRIVER   %NEP1:[TRENDS.XV15]    ! NEP1: Neptune 1 disk drive
        703%DOC      %NEP1:[TRENDS]
        703%DATA     %NEP3:[DB703]          ! NEP3: Neptune 3 disk drive
```

INDEX

--- T ---

--- U ---

--- V ---

--- W ---

--- X ---

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM-101025	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle TRENDS: The Aeronautical Post-Test Database Management System		5. Report Date January 1990
		6. Performing Organization Code
7. Author(s) W. S. Bjorkman (Analytical Mechanics Associates, Inc., Mountain View, California) and M. J. Bondi		8. Performing Organization Report No. A-88225
		10. Work Unit No. 505-61-51
9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035-1000		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001		14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: M. J. Bondi, Ames Research Center, MS 237-5, Moffett Field, CA 94035-1000
(415) 604-6341 or FTS 464-6341

16. Abstract

TRENDS, an engineering-test database operating system developed by NASA to support rotorcraft flight tests, is described. Capabilities and characteristics of the system are presented, with examples of its use in recalling and analyzing rotorcraft flight-test data from a TRENDS database. The importance of system user-friendliness in gaining users' acceptance is stressed, as is the importance of integrating supporting narrative data with numerical data in engineering-test databases.

Considerations relevant to the creation and maintenance of flight-test databases are discussed and TRENDS' solutions to database management problems are described.

Requirements, constraints, and other considerations which led to the system's configuration are discussed and some of the lessons learned during TRENDS' development are presented. Potential applications of TRENDS to a wide range of aeronautical and other engineering tests are pointed out.

17. Key Words (Suggested by Author(s)) Flight test Rotorcraft Engineering-test database Database management	18. Distribution Statement Unclassified-Unlimited Subject Category - 09

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 131	22. Price A07

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, Virginia 22161